Australian Government
Department of Defence
Defence Science and
Technology Group

Large Eddy Simulations using oodlesDST

D.A. Jones[1], C. Fureby[3], D. Norrison[1], C. Troeng[3] and R. Widjaja[2]

[1]Maritime Division, DST Group
[2]Aerospace Division, DST Group
Defence Science and Technology Group

[3]Defense Security Systems Technology,
Swedish Defence Research Agency

DST-Group-TR-3205

ABSTRACT

The oodlesDST code is based on OpenFOAM software and performs Large Eddy Simulations of turbulent fluid flows. The code contains a number of specialized subgrid scale turbulence models and wall models. This report describes these models in sufficient detail to allow a new user of the code to make informed decisions regarding the use of these models for particular applications. An overview of the code structure is given and typical solver settings and discretization choices have been illustrated by applying the code to the separation and reattachment of a turbulent boundary layer on a three-dimensional curved ramp. The simulated results are compared with experimental results as well as previous numerical simulations.

Published by

Maritime Division
Defence Science and Technology Group
506 Lorimer St
Fishermans Bend, Victoria 3207 Australia

Telephone: 1300 333 362
Fax: (03) 9626 7999

AR-016-503
January 2016

APPROVED FOR PUBLIC RELEASE

Large Eddy Simulations using oodlesDST

Executive Summary

The oodlesDST code is based on OpenFOAM software and performs Large Eddy Simulations (LES) of turbulent fluid flows. It was developed by the Hydrodynamics Group of the Swedish Defence Research Agency FOI and is made available to the Hydroacoustics Group, MD, DST Group under a Project Arrangement (PA) between the Government of the Kingdom of Sweden and the Department of Defence of Australia. The objectives of this PA include the transfer of the FOI submarine hydrodynamic modelling tools to the DST Group and the development and enhancement of the software by the Hydroacoustics Group. The oodlesDST code forms part of this technology transfer. It provides the DST Group with an enhanced ability to simulate submarine manoeuvres and provides a basis for the simulation of acoustic signatures. This report has been written to consolidate the knowledge gained from training sessions conducted by FOI in the use of the code and to provide a reference for the application of the software to a representative test case of relevance to future simulations.

The report begins with a brief introduction to the LES approach. The code contains a number of specialized subgrid scale (SGS) turbulence models and wall models. These models are described in sufficient detail to allow a new user of the code to make informed decisions regarding the use of these models for particular applications. An overview of the code structure is also given and typical solver settings and discretization choices have been illustrated by applying the code to the separation and reattachment of a turbulent boundary layer on a three-dimensional curved ramp. The behaviour of each of the subgrid scale models is described in detail and compared with both experimental results and previous numerical simulations.

Of the six SGS models studied, the Localised Dynamic Kinetic Energy Model gave the best agreement with experiment for the equilibrium boundary layer upstream of the curved ramp. In the region of separated flow the One Equation Eddy Viscosity Model with the addition of the Bardina term gave the best agreement with experiment for the positions of the separation and reattachment points, while both the Implicit LES and the Wall Adapting Local Eddy Viscosity models were best at capturing the region of reversed flow.

This page is intentionally blank

Authors

David Jones
Maritime Division

David Jones obtained a B.Sc. (Hons) and Ph.D. in Theoretical Physics from Monash University and joined Materials Research Laboratories in 1983 after positions at the University of Strathclyde, Glasgow; Queen Mary College, London University; and the University of New South Wales, Sydney. His early research covered polymer dynamics, chaos theory, atomic physics and laser-plasma interactions. During 1987/88 he was attached to the Naval Research Laboratory, Washington, DC. He has used computational fluid dynamics techniques to simulate air blast events and detonation phenomena and since moving into Maritime Division he has been simulating flow around maritime platforms using a variety of simulation techniques. He is currently using OpenFOAM software to perform both Reynolds Averaged Navier-Stokes and Large Eddy Simulations of flow around submarines.

Ronny Widjaja
Aerospace Division

Ronny Widjaja graduated from RMIT University in 2001 with a Bachelor degree in Aerospace Engineering. He was awarded a Doctor of Philosophy degree from the University of Melbourne based on his research study in Computational Aero-Acoustics. In 2005, he joined AeroStructures Pty Ltd as an aeronautical consultant and developed a CFD model for the P-3C Orion aircraft. Between 2006 and 2011, he worked for the Maritime Platforms Division of the Defence Science and Technology Organisation. His main role was to lead the submarine manoeuvring research within the Hydrodynamics Group. He worked on submarine propeller modelling during this period and completed a CRC project on advanced composite propellers. In 2012, he moved to the Air Operations Division where he applies CFD tools/capabilities to support ADF clients in helicopter aerodynamics. His areas of research include ship-helicopter operation, concurrent helicopter operation, helicopter brown-out and store release.

Daniel Norrison
Maritime Division

Daniel Norrison obtained a Bachelor of Engineering (Aerospace Engineering) and Bachelor of Applied Science (Mathematics) from RMIT University in 2004 before joining the Defence Science and Technology Organisation in 2009. He completed a Ph.D. in 2009 at RMIT University in the area of methods to accelerate grid generation and computational fluid dynamics computations for computational aeroelasticity problems. He is currently a Research Scientist in the Maritime Division where he conducts modelling and prediction of the hydrodynamics of maritime vehicles and their propulsion systems.

Christer Fureby
Swedish Defence Research Agency FOI

Christer Fureby received a M.Sc. in civil engineering at Lund University of Technology in 1989 and a Ph.D. in engineering and combustion physics from Lund in 1995. He then worked as a post-doctoral fellow in the Mechanical Engineering Department at Imperial College, London University. He has been working at FOI since 1997, first as a researcher, then as a senior researcher and since 2003 as research director in the field of computational fluid dynamics and combustion. Dr. Fureby is the head of the Computational Fluid Dynamics and Combustion group at FOI which conducts applied research in the areas of hydrodynamics and acoustics, reactive flows and contaminant transport.

Carl Troeng
Swedish Defence Research Agency FOI

Carl studied mathematics, physics and numerical analysis at Lund University and Stockholm University. After graduating as a Master of Science, he has worked at FOI since 2008. He has worked with hydrodynamic simulations, using computational fluid dynamics. The work has involved developing and maintaining in-house code within the OpenFOAM framework. During his time at FOI he has worked continuously with submarine propeller modelling.

Contents

Glossary of Terms

LES	Large Eddy Simulation
RANS	Reynolds Averaged Navier-Stokes
SGS	Sub Grid Scale
SMG	Smagorinsky Model
WALE	Wall Adapting Local Eddy Viscosity Model
OEEVM	One Equation Eddy Viscosity Model
MM	Mixed Models
DSGSM	Differential Stress Equation Model
LDKM	Localized Dynamic Kinetic Energy Model
ILES	Implicit Large Eddy Simulation

1. Introduction

In July 2010 the Hydrodynamics Group in Maritime Division (MD), DST Group, commenced a formal collaborative agreement with the Hydrodynamics Group of the Swedish Defence Research Agency FOI. The Project Agreement (PA) was entered into under the Memorandum of Understanding on Capability Development and Defence Materiel Co-operation between the Government of the Kingdom of Sweden and the Department of Defence of Australia concerning Co-operation in the Field of Research and Technology. One of the objectives of this PA is the transfer of the FOI submarine hydrodynamic modelling tools to the DST Group. These tools are implemented using the OpenFOAM software suite. One of the advantages of OpenFOAM over commercial Computational Fluid Dynamics codes is that it runs in parallel using standard MPI (Message Passing Interface) software, which means that users are not constrained by a limited number of parallel licenses and can take full advantage of available cluster hardware. Since OpenFOAM is an open source CFD software package users also have access to the source code and have complete freedom to customize the code to their own requirements.

The basic program, known as oodlesDST (object oriented large eddy simulation), solves the Large Eddy Simulation (LES) equations for incompressible flow. The program contains significant differences to the generic LES solver (oodles) available in the public OpenFOAM release, particularly in the area of the available Sub Grid Scale (SGS) models and the wall-layer models. The first part of this report provides an introduction to LES modelling and a detailed description of each of the included SGS models and the wall-models, followed by an overview of the code itself. The application of the code to a generic test case (separation and reattachment on a three-dimensional curved ramp) is then described in some detail to illustrate the FOI approach to LES simulations, to highlight the strengths and weaknesses of the SGS models, and to illustrate the effect of the numerical solution scheme on the accuracy of the simulated results.

The nature of the LES simulation approach lends itself naturally to the calculation of the turbulence generated sound spectrum. Additional software tools are required to turn the results of the simulation into power spectra and these tools are included in the software suite provided by FOI. The application of these tools to the simulation of the radiated spectra for some generic cases will be described in a forthcoming report.

2. Large Eddy Simulation

Turbulent flows are characterized by velocity fields which fluctuate rapidly in both space and time. Since these fluctuations occur over several orders of magnitude it is very expensive to construct a computational mesh which can be used to directly simulate both the small scale and high frequency fluctuations for problems of practical engineering significance. One method which can be used to eliminate the need to resolve these small

scales and high frequencies is known as Reynolds Averaging. In this approach all flow variables are divided into a mean component (which can vary slowly in time) and a rapidly fluctuating component. The Navier-Stokes equations are then averaged to remove the rapidly fluctuating components. The resulting equations, known as the Reynolds Averaged Navier-Stokes (RANS) equations, contain terms which involve mean values of products of rapidly varying quantities. These terms are known as the Reynolds Stresses and the solution of the RANS equations initially involves the construction of suitable models to represent these Reynolds Stresses.

An alternative to Reynolds averaging is Large Eddy Simulation (LES). In this approach the Navier-Stokes equations are solved without any approximations down to a specified length scale with the smaller unresolved scales modelled in the momentum equation in the form of a residual, or Sub Grid Scale (SGS), stress tensor. While the SGS stress tensor requires modelling, as do the Reynolds Stresses in the RANS approach, the advantage of the LES method is that the large scale eddies in the flow are simulated accurately. This is a more accurate approach to the simulation of turbulent flow because the large scale eddies typically contain the majority of the kinetic energy in the flow and are more responsive to flow geometry, while the smaller scale motions are more universal in nature and more easily modelled.

The LES approach was first proposed by Smagorinsky [1] in the context of meteorological flows and was first formulated in terms of a filtering operation to remove the small scale motion by Leonard [2]. In the standard approach to the derivation of the LES equations the separation into small and large scales is affected by spatially filtering the variables with a kernel function:

$$\bar{\mathrm{f}}(x_i) = \int \mathrm{f}(x_i') G(x_i, x_i')\, dx_i' \tag{1}$$

where G is the filter function and the integral is extended over the entire domain. Hence $\bar{f}(x_i)$represents a filtered version of $f(x_i)$ in which fluctuations on a scale less than the filter width Δ are smoothed out. In a CFD simulation the filtering is provided by the finite size of the computational cells. Pope [3] describes this filtering operation in more detail and gives several examples of appropriate filter functions. If this filtering is applied to the incompressible Navier-Stokes equations the following equations are obtained:

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0 \tag{2}$$

$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{\partial}{\partial x_j}\left(\bar{p}\delta_{ij} + \tau_{ij}\right) + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j \partial x_j} \tag{3}$$

where $\bar{u}_i(x_i, t)$ is the resolved velocity field, $\bar{p}(x_i, t)$ the resolved pressure field, δ_{ij} is the Kronecker delta and ν is the kinematic viscosity. The SGS stress tensor τ_{ij} is defined as:

$$\tau_{ij} = \overline{u_i u_j} - \overline{u}_i \overline{u}_j \tag{4}$$

and represents the effects of the unresolved small scales on the resolved scales. The central problem in LES is to find a suitable expression for τ_{ij}. A large number of different models have been described in the literature since Smagorinsky's original work in this field. Those contained in the oodlesDST code are described in the next section.

A more careful analysis, taking into account the fact that filtering and differentiation do not commute on realistic finite volume meshes, results in additional commutation error terms appearing in equation (3). The approach taken here is that these terms are of much smaller magnitude than τ_{ij}, hence they are either neglected, or absorbed into the model used to describe τ_{ij}.

The reader should note that the definition of τ_{ij} given in equation (4) is the one most generally used in the LES community and results in the appearance of a negative sign when τ_{ij} is defined in terms of the strain rate tensor of the filtered velocities (for example, equation (5) below). However some authors, for example Nicoud and Ducros [4], Menter [5], Lesieur et al. [6], Oran and Boris [7], Davidson [8] define the Subgrid scale stress tensor as $\tau_{ij} = \overline{u}_i \overline{u}_j - \overline{u_i u_j}$ and in this case there is no need for a negative sign in such expressions.

3. Sub Grid Scale models

3.1 The Smagorinsky Model (SMG)

The simplest SGS model is the one originally proposed by Smagorinsky [1], in which the sub-grid scale stresses are computed using an isotropic eddy viscosity approach:

$$\tau_{ij} - \tfrac{1}{3}\tau_{kk}\delta_{ij} = -2\nu_T \overline{S_{ij}} \tag{5}$$

Equation (5) defines the deviatoric part of τ_{ij} to be equal to the product of an eddy-viscosity ν_T and the resolved rate of strain tensor $\overline{S}_{ij}$, where

$$\overline{S}_{ij} = \tfrac{1}{2}\left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right) \tag{6}$$

For incompressible flow only the deviatoric part of τ_{ij} is relevant, hence it has become standard practice to deduct the trace term from the tensor and absorb it into an effective pressure field. The eddy viscosity is then calculated from an algebraic expression

involving the product of a model constant C_S, the modulus of the rate of strain tensor $|\overline{S}|$, and an expression involving the filter width Δ:

$$\nu_T = C_S^2 \Delta^2 |\overline{S}| \tag{7}$$

The problem with this approach is that there is no single value of the constant C_S which is universally applicable to a wide range of flows. If the energy spectrum follows the Kolmogorov law then C_S can be shown to have the value 0.16. This value works well for isotropic turbulence but is too high for channel flow, where a value of 0.1 is more appropriate [9]. The problem can be overcome by using the Dynamic Smagorinsky Model (DSM) devised by Germano et al. [10] and Lilly [11]. In this procedure C_S is dynamically computed during the simulation using the information provided by the resolved fields. C_S determined in this way varies with time and space, and this allows the Smagorinsky model to cope with transitional flows and to include near-wall damping effects in a natural manner. The model is difficult to implement however and suffers from a number of problems [12, 13]. In practice C_S is found to have unacceptably rapid spatial variations during a simulation that can lead to instability in the numerical computation of the resolved fields. C_S can also become negative in some regions of the mesh, implying a negative eddy viscosity and creating further instability. These problems are usually overcome either by clipping C_S so that the total viscosity (viscous plus turbulent) remains positive, or by employing some type of averaging procedure over a suitable domain, either along streamlines or planes. In complex flow geometries where no symmetries exist this procedure becomes problematic however. Only the standard (isochoric) Smagorinsky model is included in the oodlesDST code with the model constant C_s having the value 0.16.

3.2 The Wall Adapting Local Eddy Viscosity (WALE) Model

The WALE model [4] corrects some of the deficiencies of the original Smagorinsky model by defining the eddy viscosity ν_T in terms of an invariant which is more representative of the turbulent activity in the flow. The Smagorinsky model is based on the second invariant of the symmetric part of the velocity gradient tensor. Two of the problems associated with this choice are that ν_T is only related to the strain rate of the turbulent structures and not their rotation rate, and ν_T does not have the correct physical dependency on the wall normal coordinate close to the wall. The WALE model overcomes these problems by using the traceless symmetric part of the square of the velocity gradient tensor to form ν_T:

$$\nu_T = (C_w \Delta)^2 \frac{\left(S_{ij}^d S_{ij}^d\right)^{3/2}}{\left(\overline{S}_{ij}\overline{S}_{ij}\right)^{5/2} + \left(S_{ij}^d S_{ij}^d\right)^{5/4}} \tag{8}$$

$$\text{where } S_{ij}^d = \tfrac{1}{2}\left(\overline{g}_{ij}^2 + \overline{g}_{ji}^2\right) - \tfrac{1}{3}\delta_{ij}\overline{g}_{kk}^2, \quad \overline{g}_{ij}^2 = \overline{g}_{ik}\overline{g}_{kj} \text{ and } \overline{g}_{ij} = \partial \overline{u}_i / \partial x_j \tag{9}$$

The advantages of equation (8) are:

- ν_T consists of a mixture of both the local strain rate and the rotation rate, thus allowing the model to detect all the turbulent structures relevant to the kinetic energy dissipation.
- ν_T automatically goes to zero near a wall with the proper y^3 scaling.
- ν_T is zero for the case of pure shear.
- The model is numerically well conditioned; ν_T can be neither negative nor infinite.

The constant C_w has the fixed value 0.325 and, unlike the Smagorinsky model, the value is independent of the particular type of flow. The absence of a turbulent viscosity for wall bounded laminar flow implies that the development of linearly unstable waves should be possible, and in fact the model has been shown to be capable of reproducing the laminar to turbulent transition in pipe flow [4].

3.3 The One Equation Eddy Viscosity Model (OEEVM)

Both the Smagorinsky and WALE models assume that the production and dissipation of turbulent kinetic energy are in equilibrium in order to calculate the value of the independent constant (C_S or C_w) in their expressions for the turbulent viscosity. This assumption will not be true in boundary layers or for separating and re-attaching flows. To overcome this limitation the eddy viscosity can be written in the form

$$\nu_T = C_k \Delta k^{1/2} \tag{10}$$

where k is the SGS kinetic energy, which is defined in terms of the trace of the SGS stress tensor as

$$k = \tfrac{1}{2}\tau_{ii} = \tfrac{1}{2}\left(\overline{u_u^2} - \overline{u}_{ii}^2\right) \tag{11}$$

An equation for k can then be derived by taking the trace of the transport equation for the SGS stress tensor and modelling the diffusion and dissipation terms, which results in the following equation:

$$\frac{\partial k}{\partial t} + \overline{u}_j \frac{\partial k}{\partial x_j} = C_k \Delta k^{1/2} \overline{S}_{ij} \overline{S}_{ij} - C_e k^{3/2}/\Delta + \frac{\partial}{\partial x_j}\left(C_k \Delta k^{1/2} \frac{\partial k}{\partial x_j} \right) \tag{12}$$

The One Equation Eddy Viscosity Model was first proposed by Schumann [14] and later modified by Yoshizawa [15]. Fureby et al. [16] have noted that if production equals dissipation in equation (12) then the model reduces to the standard Smagorinsky Subgrid scale eddy viscosity model. The constants C_k and C_e can be obtained by assuming a $k^{-5/3}$ inertial sub-range behaviour, which gives $C_k \sim 0.07$ and $C_e \sim 1.03$. Furbey et al. [16] also comment that the strongly anisotropic behaviour of the flow close to walls and the break

down of the assumptions regarding the inertial sub-range behaviour in this region means that the OEEVM is often used with wall models.

3.4 Mixed Models (MM)

A Mixed Model refers to a combination of the scale similarity model, first introduced by Bardina et al. [17], with any form of eddy viscosity model. The Bardina model is based on the concept of scale invariance and assumes that the structure of the velocity field at scales below the size of the filter width Δ is similar to that at scales above Δ. This postulate is supported by the experiments of Liu et al. [18]. Based on this assumption an approximate expression for the SGS stress tensor can be written as follows:

$$\tau_{ij} = \overline{\overline{u}_i \overline{u}_j} - \overline{\overline{u}}_i \overline{\overline{u}}_j \tag{13}$$

This assumption allows the calculation of the SGS stresses directly from the resolved fields. The second bar above the velocity components in equation (13) represents a second filter at a scale γΔ, where γ ≥ 1. The original model by Bardina used γ =1 while Liu et al. [18] used γ = 2.

That equation (13) is only an approximation to the true SGS stress tensor can easily be seen by writing the velocity in equation (4) in the form $u_i = \overline{u}_i + u_i'$ and then evaluating the expression for the SGS stress tensor:

$$\tau_{ij} = \left(\overline{\overline{u}_i \overline{u}_j} - \overline{\overline{u}}_i \overline{\overline{u}}_j\right) + \left(\overline{\overline{u}_i u_j'} + \overline{u_i' \overline{u}_j} - \overline{u_i'}\,\overline{\overline{u}}_j - \overline{\overline{u}}_i \overline{u_j'}\right) + \left(\overline{u_i' u_j'} - \overline{u_i'}\,\overline{u_j'}\right) \tag{14}$$

The first term on the right side of equation (14) represents the effects of interactions between the smallest resolved scales, the second term represents the effects of interactions between the resolved and unresolved scales and the third term is the effect of interactions amongst the unresolved scales. This last term in equation (14) is the one which represents the bulk of the SGS dissipation.

Bardina et al. [17] showed that equation (13) resulted in a high correlation between the real and modelled SGS stresses but lead to inaccurate results when used in simulations because the model was insufficiently dissipative. To account for the additional dissipative terms missing from equation (13) Bardina suggested adding a dissipative Smagorinsky term to give the following mixed model:

$$\tau_{ij} = \overline{\overline{u}_i \overline{u}_j} - \overline{\overline{u}}_i \overline{\overline{u}}_j - 2\left(C_S \Delta\right)^2 \left|\overline{\overline{S}}\right| \overline{\overline{S}}_{ij} \tag{15}$$

The choice of the Smagorinsky model is rather arbitrary however and any SGS turbulent viscosity can be used. The oodlesDST code currently contains two mixed models, the SMGMIXED model refers to a combination of the Smagorinsky model and the Bardina term, whilst the OEEVMMIXED model refers to the combination of the One Equation Eddy Viscosity model with the Bardina term.

3.5 The Localised Dynamic Kinetic Energy Model (LDKM)

The Localised Dynamic Kinetic Energy Model of Kim and Menon [19, 20] is a one equation eddy viscosity model in which the constants C_k and C_e are determined dynamically during the course of the simulation. This is done by defining a second test filter of width $\gamma\Delta$ (where typically $\gamma = 2$) and using scale similarity arguments similar to those used in the Mixed Model. A test-scale Leonard stress tensor is defined as:

$$L_{ij} = \overline{\overline{u}_i \overline{u}_j} - \overline{\overline{u}}_i \overline{\overline{u}}_j \tag{16}$$

where the second overbar again denotes filtering by the test filter. A resolved kinetic energy at the test-filter level can also be defined by:

$$k_{test} = \tfrac{1}{2}\left(\overline{\overline{u}_k \overline{u}_k} - \overline{\overline{u}}_k \overline{\overline{u}}_k\right) \tag{17}$$

which is dissipated at the small scales by:

$$e = (\nu + \nu_T)\left(\overline{\frac{\partial \overline{u}_i}{\partial x_j}\frac{\partial \overline{u}_i}{\partial x_j}} - \frac{\partial \overline{\overline{u}}_i}{\partial x_j}\frac{\partial \overline{\overline{u}}_i}{\partial x_j}\right) \tag{18}$$

Assuming that the same parameterization can be used for both τ_{ij} and L_{ij} we can write:

$$L_{ij} - \tfrac{1}{3}\delta_{ij} L_{kk} = -2C_k \overline{\Delta}\, k_{test}^{1/2}\, \overline{\overline{S}}_{ij} \tag{19}$$

where the overbar on Δ indicates that the width of the test filter is to be used. Using the similarity assumption an over-determined set of equations for C_k can then be found which are solved using the least-squares method suggested by Lilly [11] to give:

$$C_k = \tfrac{1}{2}\frac{L_{ij}\sigma_{ij}}{\sigma_{ij}\sigma_{ij}} \tag{20}$$

where:

$$\sigma_{ij} = -\overline{\Delta}\, k_{test}^{1/2}\, \overline{\overline{S}}_{\mathrm{ij}} \tag{21}$$

Similarity between the dissipation rate at the grid-filter level and the test-filter level allows the following expression for the dissipation model coefficient to be obtained:

$$C_e = (\nu + \nu_T)\overline{\Delta}\left(\overline{\frac{\partial \overline{u}_i}{\partial x_j}\frac{\partial \overline{u}_i}{\partial x_j}} - \frac{\partial \overline{\overline{u}}_i}{\partial x_j}\frac{\partial \overline{\overline{u}}_i}{\partial x_j}\right) / k_{test}^{3/2} \tag{22}$$

The denominators in the expressions for C_k and C_e are well defined and the ill-conditioning problem associated with the dynamic model based on Germano's approach [10] is avoided with these expressions. The above equations have been shown to enable

stable localized evaluations of the model coefficients without employing any special spatial or temporal averaging.

3.6 Differential Stress Equation Model (DSGSM)

One of the weaknesses in all of the sub grid scale models described so far has been the use of the isotropic eddy viscosity assumption, equation (5) in Section 3.1. This assumption implies that the principal axes of τ_{ij} are coincident with those of the mean strain rate tensor. There are many flows for which this assumption is incorrect, including flow over curved surfaces and flow with boundary layer separation, flows which are particularly pertinent to maritime platforms. For RANS simulations this assumption can be avoided by using a Reynolds-stress model [3], in which model transport equations are solved for each of individual Reynolds stresses $\langle u_i u_j \rangle$ and for the dissipation ε. A similar approach can be applied to LES by forming an equation for the SGS stress tensor τ_{ij}. Deardorff [21] was the first to use this approach. He derived sub grid transport equations for each of the sub grid Reynolds stresses and fluxes, rather than solving a single equation for a sub grid eddy viscosity. He applied this method to model atmospheric boundary layers and found superior results with this approach, but also noted that the method was computationally more expensive than the less rigorous eddy viscosity methods.

More recently Fureby et al [22] have followed the approach pioneered by Deardorff [21] and studied the sub grid scale stress equation using the principles of frame indifference and the concept of realizability. Closure models were proposed which satisfied these constraints and satisfied the additional requirement that the modelled balance equation must degenerate into the ordinary balance equation for the sub grid scale kinetic energy when contracted. The model was applied to forced homogeneous isotropic turbulence and was found to reproduce the interscale energy transfer better than traditional sub grid scale models for a larger range of Reynolds numbers. When applied to fully developed turbulent channel flow the model significantly improved the first and second order velocity moments and predicted the mean streak spacing better than any other model. The additional computational cost was found to be no more than about 20% of that used for the constant coefficient Smagorinsky model [22].

In this section we describe the differential stress equation model as implemented in oodlesDST. The presentation is an abbreviated form of that recently given by Fureby et al. [16]. An exact equation for τ_{ij} can be obtained by combining the original and filtered forms of the Navier-Stokes equations. In symbolic form this can be written as:

$$\frac{\partial \tau_{ij}}{\partial t} + \frac{\partial}{\partial x_k}\left(\tau_{ij}\overline{u}_k\right) = \mathrm{P}_{ij} + \mathrm{J}_{ij} + \Phi_{ij} + \mathrm{E}_{ij} \qquad (23)$$

The term P_{ij} represents the rate of creation of τ_{ij} by the resolved strain and has the following exact form:

$$P_{ij} = -\tau_{ik}\frac{\partial \overline{u}_j}{\partial x_k} - \tau_{jk}\frac{\partial \overline{u}_i}{\partial x_k} \tag{24}$$

J_{ij} is a diffusive term and promotes spatial redistribution of τ_{ij}. It is modelled using a conventional gradient law using the subgrid viscosity $\nu_k = c_k \Delta k^{1/2}$ where c_k is a model coefficient:

$$J_{ij} = -\frac{\partial}{\partial x_k}\left(\nu_k \frac{\partial \tau_{ij}}{\partial x_k}\right) \tag{25}$$

Φ_{ij} represents the correlation between pressure and strain and serves to redistribute energy among the normal subgrid stress components. It can be divided into three independent contributions: a term responsible for non-linear interactions, a term representing the return to isotropy by the velocity gradients, and a term representing wall-reflection effects due to the pressure fluctuations. Only the first two terms are modelled in oodlesDST and so Φ_{ij} takes the following form:

$$\Phi_{ij} \approx c_m \frac{k^{1/2}}{\Delta}\left(\tau_{ij} - \frac{2}{3}k\delta_{ij}\right) + \frac{2}{5}k\left(\overline{S}_{ij} - \frac{1}{3}\overline{S}_{kk}\delta_{ij}\right) \tag{26}$$

E_{ij} represents the rate of destruction of τ_{ij} by viscous action. It arises from the fine scale motion because the velocity gradients are steepest at these scales. For a high Reynolds number these motions are assumed to be virtually isotropic, so that E_{ij} can be written as:

$$E_{ij} \approx c_e \frac{k^{3/2}}{\Delta}\delta_{ij} \tag{27}$$

where c_e is a second model coefficient. Combining equations (23) through (27) the full model than becomes:

$$\frac{\partial \tau_{ij}}{\partial t} + \frac{\partial}{\partial x_k}\left(\tau_{ij}\overline{u}_k\right) - \frac{\partial}{\partial x_k}\left(\nu_k \frac{\partial \tau_{ij}}{\partial x_k}\right) = -\tau_{ik}\frac{\partial \overline{u}_j}{\partial x_k} - \tau_{jk}\frac{\partial \overline{u}_i}{\partial x_k} - c_m \frac{k^{1/2}}{\Delta}\tau_{ij} + \frac{2}{5}k\overline{S}_D + \left(\frac{2}{3}c_m - c_e\right)\frac{k^{3/2}}{\Delta}\delta_{ij} \tag{28}$$

where $\overline{S}_D$ represents the deviatoric part of $\overline{S}$ and $c_e = 0.69$, $c_m = 4.13$ and $c_k = 0.07$. Fureby et al [16] note that when the trace of equation (28) is taken the resulting equation is identical to the transport equation for k used in the OEEVM.

Equation (28) is implemented in oodlesDST but has so far not been tested on many canonical flows. Apart from the earlier applications to forced homogeneous isotropic turbulence and fully developed turbulent channel flow [22] it has been used to study the flow around a 6:1 prolate spheroid and to simulate the three-dimensional contoured ramp problem of Song et al. [23]. In these latter applications the success of the model has been mixed; improved predictions were noted for the flow around the 6:1 prolate spheroid;

however for the contoured ramp problem the simulated results were inferior to those calculated using the LDKM model.

3.7 Implicit LES (ILES)

The idea behind Implicit LES is to solve the unfiltered Navier-Stokes equations without using an explicit SGS model and instead use the inherent dissipation contained in the discretization scheme to act as an implicit SGS model. This approach was pioneered by Boris [24] at the Naval Research Laboratories and was originally known as MILES (Monotone Integrated Large Eddy Simulation). Boris and Book [25] had previously developed the Flux-Corrected Transport (FCT) algorithm in order to numerically simulate the behaviour of strong shocks in compressible flows. FCT is a non-linear monotone algorithm which has been remarkably successful in time-dependent simulations of shear flows, jet flows, compressible turbulence, and Rayleigh-Taylor mixing. Combining the experience gained from solving many compressible flow problems using the FCT algorithm with those of other researchers using related monotone algorithms, Boris et al. [26] conjectured that monotone algorithms can actually be viewed as LES models with an intrinsic SGS model.

This approach has been studied in detail by Fureby and Grinstein [27] and Grinstein and Fureby [28] and applied to isotropic decaying turbulence, transitional jets, channel flows, free shear flows, flow over a prolate spheroid and flow around submarines. The overwhelming conclusion from all these studies is that MILES is capable of predicting the experimental results equally as well as any of the other standard LES approaches. An excellent summary of recent work in this area is contained in the book by Grinstein et al. [29].

One advantage of the MILES approach is that, as with the Differential Stress Equation SGS model of the previous section, it avoids the use of the isotropic eddy viscosity assumption. Fureby and Grinstein [27] have shown that MILES reproduces the first-order and second-order statistical moments of the velocity field for wall-bounded flows almost as accurately as the DSGSM and better than isotropic eddy-viscosity models. Through the use of the modified equation analysis technique they have been able to show that, for a particular class of flux-limiting monotone schemes, this is attributable to the fact that MILES is equivalent to a non-linear tensorial eddy viscosity model.

Not all discretization schemes reproduce the correct physics of the energy cascade however. In reality, energy is removed from the flow at the Kolmogorov scale by the physical viscosity. In an LES simulation the mesh width is several orders of magnitude larger than the Kolmogorov scale and without a SGS model the grid scale fluctuations in the energy build up to nonphysical levels because the excitation cannot be removed at the rate it is generated. Conventional CFD algorithms, such as the upwind scheme for example, remove energy at all wavelengths, which is clearly nonphysical. In contrast to this, nonlinear monotone schemes remove energy only at wavenumbers close to the filter width, thereby reproducing the effect of an explicit SGS model.

The choice of suitable schemes is discussed in detail in the book by Grinstein et al. [29]. As described by Fureby and Grinstein [27], the methods available for constructing implicit SGS models are restricted to expressing the convective fluxes at the cell faces in terms of high-resolution non-linear algorithms. This implies that the methods will be at least second-order accurate on smooth solutions yet still give well resolved, nonoscillatory solutions at sharp gradients and discontinuities. The velocity at a cell face is written in the following form

$$\mathbf{v}_f = \mathbf{v}_f^H - (1-\Gamma)(\mathbf{v}_f^H - \mathbf{v}_f^L) \tag{29}$$

where $\mathbf{v}_f^H$ represents a high-order convective flux function obtained from either linear or cubic interpolation resulting in second order or fourth order schemes respectively, while $\mathbf{v}_f^L$ is a low order dispersion free convective flux function obtained from an upwind biased piecewise constant approximation. The flux limiter Γ has to be constructed so as to allow as much as possible of the correction or anti-diffusion term $(\mathbf{v}_f^H - \mathbf{v}_f^L)$ to be included without increasing the variation of the solution, so as to comply with the physical principles of causality, monotonicity and positivity (where appropriate) and hence preserve the properties of the Navier-Stokes equation. Experience has shown that most of the monotonicity preserving schemes such as FCT [25], the Gamma limiter [30] and the Monotonized Central (MC) scheme [31] work well for ILES. The oodlesDST code currently contains both the Gamma and MC schemes (amongst many others), but not the FCT scheme.

4. Wall models

Flow in a turbulent boundary layer is highly non-isotropic and contains a number of coherent structures, including high and low velocity streaks and hairpin vortices. A good introduction to the complicated nature of these flows is given by Davidson [8] and by de Villiers [32]. Resolution of these flow features in a numerical simulation requires a very fine mesh. Bagget [33] suggests that a wall-resolved LES requires a mesh satisfying the following constraints: $\Delta x^+ < 100$, $\Delta y^+ < 2$, $\Delta z^+ < 20$, where x, y and z are the streamwise, wall normal and spanwise directions and where lengths are given in wall units, defined by $x^+ = x u_\tau / \nu$ where u_τ is the friction velocity defined by $u_\tau = (\tau_w/\rho)^{1/2}$ and τ_w is the wall shear stress. Sagaut [12] however states that the above criteria would lead to a poor resolution LES with unphysical streaks and large errors in the skin friction. His recommendation for high resolution LES is: $\Delta x^+ < 50$, $\Delta z^+ < 12$, with three mesh points located in the region $0 \le y^+ \le 10$. Menter [5] is even stricter and recommends $\Delta x^+ < 30$ - 40, $\Delta y^+ \sim 1$, $\Delta z^+ < 15$ – 20. Whilst these recommendations can be met when simulating model experiments at moderate Reynolds numbers, they are totally impractical for the simulation of a full scale submarine undergoing a manoeuvre, for example. For these types of simulations, wall models and a coarse mesh adjacent to the wall are required.

Fureby et al. [34] and Fureby [35] have developed a wall model suitable for the simulation of high Reynolds number wall bounded flows which takes into account the influence of a streamwise pressure gradient and have successfully used this model to simulate fully developed channel flow and flow over a 3D surface mounted hill [36], as well as more challenging simulations such as self-propelled submarines at straight course [37]. The development of this model starts with the thin-boundary layer equations in the following form [38]:

$$\frac{\partial \overline{u}_i}{\partial t} + \frac{\partial}{\partial x_j}\left(\overline{u}_i \overline{u}_j\right) + \frac{\partial \overline{p}}{\partial x_i} = \frac{\partial}{\partial y}\left(\nu \frac{\partial \overline{u}_i}{\partial y} - \tau_{ij}\right) \tag{29}$$

where $\overline{u}_i$ is the velocity in the x_i direction and y indicates the wall normal direction. This equation may be integrated in the wall normal direction assuming that the convective terms are negligible in comparison to the other terms close to the wall and that the streamwise pressure gradient is constant in the wall normal direction. This results in

$$y \frac{\partial \overline{p}}{\partial x} = \nu \frac{\partial \overline{u}_x}{\partial y} - \tau_{xy} - \tau_w \tag{30}$$

where τ_w is the wall shear stress defined by $\tau_w = \nu \left.\frac{\partial \overline{u}_x}{\partial y}\right|_{y=0}$. Equation (30) can be further simplified in two special cases. In the viscous region very close to the wall the subgrid stresses can be neglected so that equation (30) becomes

$$y \frac{\partial \overline{p}}{\partial x} = \nu \frac{\partial \overline{u}_x}{\partial y} - \tau_w \tag{31}$$

whereas outside the viscous region the subgrid stresses dominate and equation (30) becomes

$$y \frac{\partial \overline{p}}{\partial x} = -\tau_{xy} - \tau_w \tag{32}$$

Each of these equations can be integrated one more time in the wall normal direction to give an expression for the mean streamwise velocity in the appropriate region. Before doing so however the equations are scaled using the extended inner scaling proposed by Manhart et al. [39]. This scaling takes into account both the wall shear stress and the pressure gradient and the scaling is denoted by a superscript *. We define $u_p = \left|\nu \frac{\partial \overline{p}}{\partial x}\right|^{1/3}$, $u_{\tau p} = \left(\tau_w + u_p^2\right)^{1/2}$, $u_\tau = \tau_w^{1/2}$ and $\alpha = u_\tau^2 / u_{\tau p}^2$ the non-dimensional velocity $\overline{u}_x^*$ and length y^* are defined by $\overline{u}_x^* = \overline{u}_x / u_{\tau p}$ and $y^* = u_{\tau p} y / \nu$.

In terms of these scaled variables equation (31) can then be integrated in the wall normal direction to give

$$\bar{u}_x^* = \alpha y^* + \tfrac{1}{2}(1-\alpha)^{3/2}(y^*)^2 \qquad \text{if } y^* < y_C^* \tag{33}$$

where y_C^* is the solution to the equation

$$\left(\alpha - \kappa^{-1}[1-\alpha]^{3/2}\right)y^* + \tfrac{1}{2}[1-\alpha]^{3/2}(y^*)^2 - \kappa^{-1}ln(y^*) - B = 0 \tag{34}$$

and $\kappa \approx 0.41$ and $B \approx 5.3$ in a zero pressure gradient flow.

The non-dimensional parameter α is used to quantify the relative effects of shear stress and streamwise pressure gradient; $\alpha = 0$ corresponds to a zero shear stress flow and $\alpha = 1$ corresponds to a zero pressure gradient flow.

To integrate equation (32) one more time a simple subgrid model for τ_{xy} is used, $\tau_{xy} = -\nu_k \frac{\partial}{\partial y}(\bar{u}_x)$, where $\nu_k = \kappa u_{\tau p} y$. The equation can then be integrated in the wall normal direction to give the following expression for $\bar{u}_x^*$ in the constant stress region outside of the viscous region

$$\bar{u}_x^* = \kappa^{-1}ln(y^*) + \kappa^{-1}\tfrac{1}{2}(1-\alpha)^{3/2}(y^*)^2 + B \qquad \text{if } y^* \geq y_C^* \tag{35}$$

Although equations (33) and (35) take into account the effects of a mild pressure gradient and work well in the viscous sub-layer ($y^* < 5$) and in the logarithmic region ($30 < y^* < 1000$) they are less effective in the intermediate buffer region ($5 < y^* < 30$). For this reason, and also because it has been found that the switching involved in the use of equations (33) and (35) in a CFD code causes numerical instabilities, it has been found preferable to use the following implicit expression for $\bar{u}_x^*$ derived by Spalding [40].

$$y^* = \bar{u}_x^* + e^{-(\kappa B)}\left\{e^{(\kappa \bar{u}_x^*)} - \left[1 + (\kappa\bar{u}_x^*) + \tfrac{1}{2!}(\kappa\bar{u}_x^*)^2 + \tfrac{1}{3!}(\kappa\bar{u}_x^*)^3 + ...\right]\right\} \tag{36}$$

Equation (36) is formally only valid for zero pressure gradient flows, however the smooth nature of the transition between the viscous, buffer and logarithmic regions contained in this expression has proven to produce better agreement between simulated and experiment results than the use of equations (33) and (35).

The implementation of the wall model described by either equations (33), (35) or (36) in oddlesDSTO is achieved by modifying the subgrid scale viscosity in the first cell adjacent to the wall to force the wall normal velocity gradient to correspond to the shear stress value given by the wall model. In these cells the effective viscosity, which is normally the sum of the molecular viscosity ν and the subgrid viscosity ν_k, is modified by adding a subgrid wall viscosity ν_{BC} to ν so that the effective viscosity ν_{eff} becomes

$$\nu_{\text{eff}} = \nu + \nu_{\text{BC}} = \tau_w \Big/ \left(\frac{\partial \overline{u}_x}{\partial y}\right)_{\text{P}} = \mathrm{u}_\tau^2 \Big/ \left(\frac{\partial \overline{u}_x}{\partial y}\right)_{\text{P}} \approx u_\tau^2 y_{\text{P}} / \overline{u}_{x,\text{P}} \qquad (37)$$

In equation (37) τ_w is calculated from either the bi-regional log law, i.e. equations (33) and (35), or from Spalding's continuous log law, equation (36). The normal velocity at the wall is evaluated at the first grid point P using the expression $\overline{u}_{x,\text{P}} / y_{\text{P}}$. Duprat et al. [41] have recently published a scheme similar to the one described above.

5. The oodlesDST Code

The oodlesDST code is based on the OpenFOAM software suite [42], which is an open source object-oriented library for numerical simulations in continuum mechanics. OpenFOAM is written in the C++ programming language and draws heavily upon object-oriented programming techniques to allow a natural representation of the underlying partial differential equations. The software comes with a number of existing solvers for standard applications, such as icoFoam, a transient solver for incompressible laminar flow, pisoFoam, a transient solver for incompressible turbulent flow, and simpleFoam, a steady-state solver for incompressible, turbulent flow. An introduction to OpenFOAM and information about the general running of the standard codes, data structures, applications, libraries and post-processing is available in the User Guide [43].

One of the advantages of OpenFOAM is that users who have the requisite knowledge of object-oriented programming techniques in the C++ language and a familiarity with the OpenFOAM class structure can write their own solvers and utilities for specific applications relatively quickly by using the pre-defined constructs in the OpenFOAM library. The oodlesDST code is an example of this approach; the code solves the LES form of the Navier-Stokes equations, i.e. equations (2) and (3) from section 2, and incorporates each of the SGS models described in section 3 and the wall models described in section 4. The code consists of the main programme, contained in the file oodlesDST.C, along with ten additional files. A brief description of the function of each of these additional files is shown in Appendix A.

In order to be able to use the code efficiently and correctly it is important to be familiar with the numerical techniques which are used to discretise the governing partial differential equations. The momentum and continuity equations form a coupled set of equations for the three components of the velocity vector and the pressure. The momentum equation can be used to solve for the velocity components if the pressure is known, while the continuity equation imposes a constraint on these velocity components. To provide an equation for pressure the continuity equation can be combined with the momentum equation to construct a Poisson equation for the pressure alone. The simplest solution procedure is to proceed iteratively, first assume that the pressure is known and solve for an approximate velocity value, then use the approximate velocity to construct the Poisson equation which is then solved for an updated value of the pressure. The loop is

then iterated several times until the continuity equation is satisfied to within an acceptable level of accuracy.

In oddlesDSTO the solution strategy described above is carried out using the PISO (Pressure Implicit Splitting of Operators) algorithm [44]. The coupled pressure-velocity system contains two complex coupling terms:

- The non-linear convection term, containing a term quadratic in the velocity.
- The linear pressure-velocity coupling.

The non-linearity in the convection term ($\nabla \cdot (\mathbf{uu})$) is handled using an iterative solution technique by making the approximation

$$\nabla \cdot (\mathbf{uu}) \approx \nabla \cdot (\mathbf{u}^{o}\mathbf{u}^{n}) \tag{38}$$

where $\mathbf{u}^{o}$ is the currently available solution and $\mathbf{u}^{n}$ is the new solution**.** The algorithm cycles until $\mathbf{u}^{o} = \mathbf{u}^{n}$. The basic assumption behind the PISO algorithm is that for small times steps (or low Courant numbers) the pressure-velocity coupling is much stronger than the non-linear coupling. Therefore it is possible to repeat a number of pressure corrections without updating the discretization of the momentum equation (i.e. without updating $\mathbf{u}^{o}$).

This algorithm is shown in more detail in Appendix B, which shows the time loop from oodlesDST.C. This loop is very similar to the time loop contained in the codes icoFoam or pisoFoam. To understand this part of the code in more detail we write the momentum equation in the following semi-discretized form:

$$a_{P}^{u}\mathbf{u}_{P} + \sum_{N} a_{N}^{u}\mathbf{u}_{N} = \mathbf{r} - \nabla p \tag{39}$$

In equation (39) the exact form of the coefficients a_{P}^{u} and a_{N}^{u} depends on the method used to discretise each of the terms on the left hand side of the Navier-Stokes equation, i.e. the time dependent term, the convection term and the diffusion term. The source term vector is represented by $\mathbf{r}$. Equation (39) is further simplified by defining the vector $\mathbf{H}$ as follows:

$$\mathbf{H}(\mathbf{u}) = \mathbf{r} - \sum_{N} a_{N}^{u}\mathbf{u}_{N} \tag{40}$$

The $\mathbf{H}$ vector is a combination of all the neighbour matrix coefficients multiplied by the velocity plus all the non-linear source terms minus the pressure gradient. An expression for the velocity $\mathbf{u}_{P}$ at node P can now be written as:

$$\mathbf{u}_{P} = (a_{P}^{u})^{-1}(\mathbf{H}(\mathbf{u}) - \nabla p) \tag{41}$$

By substituting equation (41) into the continuity equation the following Poisson equation for pressure is obtained:

$$\nabla \cdot \left(\left(a_P^u \right)^{-1} \nabla p \right) = \nabla \cdot \left(\left(a_P^u \right)^{-1} \mathbf{H}(\mathbf{u}) \right) \tag{42}$$

The time loop in oodlesDST starts with the statements

include "readPISOControls.H"

include "CourantNo.H"

These files read the PISO controls in the PISO subdictionary of the fvSolution file and then calculate the Courant number and write it to the screen. The statements

include "subGridModel.H"

include "wallViscosity.H"

contain the coding for each of the sub grid scale models described in section 3 and the wall models described in section 4. The statement

include "UEqn.H"

constructs and solves the appropriate form of the momentum equation and provides an initial estimate for the velocity $\mathbf{u}^o$ at the start of the timestep.

We then enter the PISO loop, which is controlled by the number of corrector steps, nCorrectors, which is read from the fvSolution file. First a new estimate of the velocity is obtained, rUA*UEqn.H(), which is known as the momentum predictor and represents the new velocity without the effect of the pressure gradient. It corresponds to the expression

$$\mathbf{u}_P^* = \left(a_P^u \right)^{-1} \mathbf{H}(\mathbf{u}) \tag{43}$$

Next the face fluxes Φ_f are constructed by interpolation from the approximate velocity field

$$\Phi_f = \mathbf{u}_f^* \cdot \mathbf{S}_f \tag{44}$$

and are adjusted to ensure that they are globally conservative. The statement

include "pEqn.H"

solves equation (42) for the pressure. This step is performed a number of times, depending on the number of nonorthogonal correctors (nNonOrthCorr), and at the end of the loop the flux Φ is corrected for the next pressure-corrector step. The approximate velocity field $\mathbf{u}_P^*$ is then updated using the contribution from the corrected pressure field, U = U – rUA*fvc::grad(p), corresponding to

$$\mathbf{u}_{new} = \mathbf{u}^{*} + \mathbf{u}' \tag{45}$$

where $\mathbf{u}^{*}$ is given by equation (43) and $\mathbf{u}'$ is given by $\mathbf{u}' = -a_p^{u-1}(\nabla p)$. A flow chart illustrating this algorithm is shown in Figure 1.

The fvSolution file allows the user to specify two different solvers for the pressure iterations. If there are nCorrectors then the "p" solver will be used for the first "nCorrectors-1" steps while the "pFinal" solver will be used on the last pass through the loop. The reason for this is that the first "nCorrectors-1" steps do not need to be solved with the same degree of accuracy as the final step. The intermediate steps are used to form updated expressions for the pressure equation but the updated velocity is only calculated after the final step. Hence the tolerance on the "pFinal" solver is usually set several orders of magnitude higher than the tolerance on the "p" solver. This allows the user to minimize the number of iterations during the intermediate steps and ensure maximum accuracy at the final step.

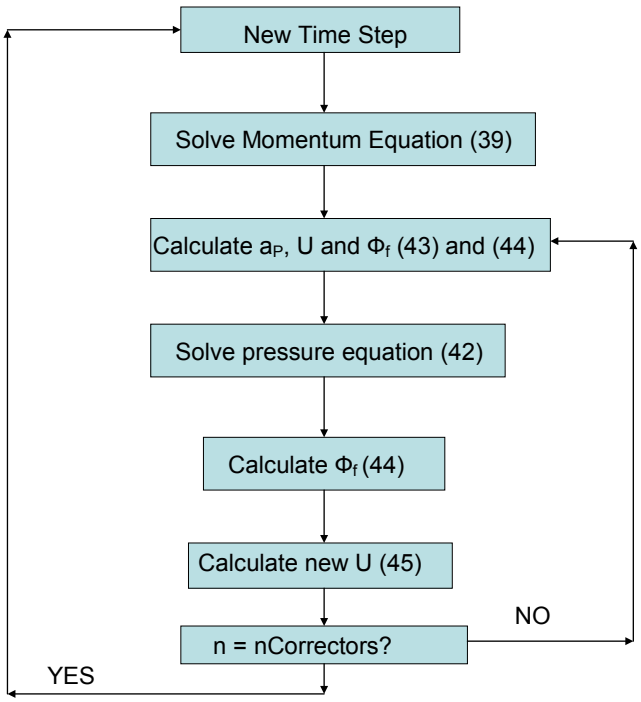

Figure 1. Flow chart for PISO algorithm

6. Flow over a curved ramp

The simulation of turbulent boundary layer separation and reattachment is a challenging problem for current CFD capabilities. Song et al. [23] have experimentally studied the separation, reattachment and then redevelopment of a turbulent boundary layer over a smoothly contoured ramp and a downstream flat plate. Their experiment has subsequently been used as a test case for a number of CFD simulations, both RANS and LES [45 – 52]. In this section we provide a description of the experiment performed by Song et al [23], discuss the results of previous LES simulations of this experiment, and then describe simulation results obtained using oodlesDST.

6.1 Experimental Description

The experiments were performed in a close-loop wind tunnel having a test section 152 mm by 711 mm by 3.0 m long. A special insert was placed into the rectangular test section to form the ramp geometry. A two-dimensional schematic of the downstream end of this insert is shown in Figure 2. In the experiment a two-dimensional boundary layer develops along the upstream flat plate and then flows down a curved ramp, where it separates and then reattaches and redevelops on the downstream flat plate. The two-dimensionality of the flow was checked by measuring the streamwise mean and fluctuating velocity at several spanwise locations. The measurements showed that the flow in the central region of the test section was spanwise uniform and was not affected by the side wall boundary layers. The ramp itself is a portion of a circular arc having a radius of 127 mm and its horizontal extent, denoted to be the ramp length L, is 70 mm. Experimental data are presented in a coordinate system in which x is the freestream direction and y is the wall normal direction. The y axis is maintained vertical and does not follow the curvature of the ramp. A normalized streamwise coordinate, $x' = (x\text{-}x_0)/L$, where x_0 corresponds to the beginning of the ramp, is also used. The flow has a free stream velocity of 20.4 m/s and the working medium is air. The freestream turbulence intensity is approximately 0.2%. The Reynolds number based on the step height of 21 mm is 28,600.

The boundary layer on the upstream plate has a thickness of approximately 25.3 mm as it approaches the ramp. The curvature of the ramp initially produces a favourable pressure gradient up to $x' = 0.16$, after which the effect of the expansion dominates and causes a strong adverse pressure gradient which leads to separation of the boundary layer at $x' = 0.77$. Reattachment occurs at $x' = 1.36$ and the horizontal extent of the separation bubble is 41 mm. The height of the separation bubble at the trailing edge is 4.7 mm. Experimental velocity measurements are available at several streamwise locations between $x/L = -2$ and $x/L = 7$.

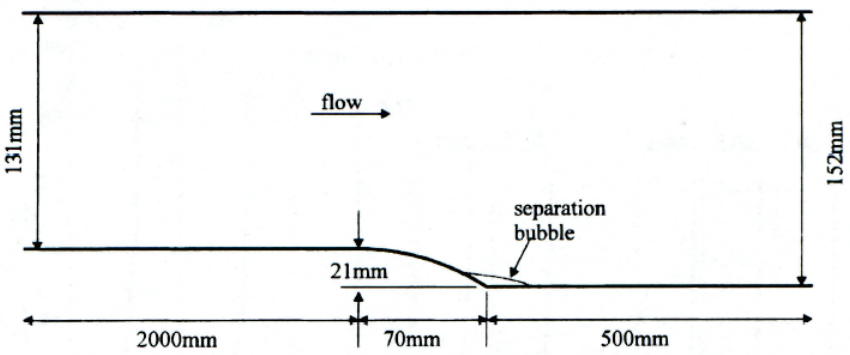

Figure 2: Two-dimensional schematic of contoured ramp experiment of Song et al [23].

6.2 Previous simulation results

Svennberg and Fureby [45] used LES with both the SMG and OEEVM subgrid scale models to simulate the experiment of Song et al. [23]. Two three-dimensional meshes were used, one with approximately 160,000 non-uniformly distributed cells and a spanwise width of 100 mm, and one wider mesh with approximately 245,000 cells and a width of 200 mm. The wall normal spacing was kept constant throughout the whole domain for both meshes and the y^+ values varied between 3 and 30. Explicit wall models were used to mimic the near wall behaviour. The structure of the flow field was found to be highly dependent on both the spanwise extent of the computational domain and the boundary conditions used at the side walls. For the mesh with the 100 mm wide domain large scale vortices were found to persist in the recirculation zone after simulation times of eight flow through times. This simulation time is normally considered to be long enough to obtain statistically converged results. When symmetry planes were used at the side walls there was one large vortex, and when periodic boundary conditions were used a pair of counter rotating vortices was observed. These vortices eventually broke up when the simulation time was extended to 15 flow through times. These semi-stable vortices were never observed however after either 7.5 or 15 flow through times when the mesh having a width of 200 mm was used.

Simulations on the wider domain were performed using both the OEEVM and SMG models. Both models gave almost the same results for the position of the separation and reattachment points and the mean velocity profiles at a representative number of axial stations were in very good agreement with the experimental results. The simulation using the OEEVM gave a separation starting at $x' = 0.66$ and reattachment occurring at $x' = 1.36$, whilst the results for the SMG model were $x' = 0.67$ and $x' = 1.29$ respectively. Based on the length of the separation bubble the results from the OEEVM are in better agreement with experiment than those from the SMG model.

More recently Feymark et al. [46] and Fureby [47] have reported LES results on structured hexahedral meshes with a 200 mm spanwise width and symmetry boundary conditions containing both 1.0 and 8.0 million cells. The sub grid scale models used were LDKM,

WALE, and OEEVMMIXED with a wall model. Best agreement with the experimental data was obtained with the LDKM model, closely followed by the OEEVMMIXED plus wall model.

Wasistho and Squires [48, 49] have performed both RANS and LES simulations on the contoured ramp experiment. For the LES simulation they used a wall resolving mesh containing approximately 2.9 million cells and the dynamic Smagorinsky model. The two-dimensional RANS simulations were performed using both the Spalart-Allmaras [53] and $\overline{v^2} - f$ [54] turbulence models on a mesh containing approximately 17,000 cells. For both the LES and RANS simulations they found fairly good agreement between the simulated results and the experiments for the mean flow. Both the RANS and LES simulations however showed the mean separation occurring too early, at approximately $x' = 0.4$, while the reattachment point was accurately simulated at $x' = 1.4$.

Radhakrishnan et al. [50] have also performed both RANS and LES simulations for this experiment. The RANS models used were the standard k-ε high Reynolds number turbulence model [55] with a two-layer approach to obtain the eddy viscosity in the low Reynolds number near-wall region, the Shear stress transport model (SST) of Menter [56] and the standard Spalart-Allmaras model [53]. The LES simulations used the Detached Eddy Simulation (DES) approach in which the Spalart-Allmaras model was used to compute the eddy viscosity in the near-wall region while the standard LES equations were solved away from the wall. The LES simulations were performed on both a coarse and fine mesh containing 1.5 million and 5.7 million cells respectively and the first grid point in the wall-normal direction was located at y^+=1. The RANS simulations were performed on two-dimensional meshes similar to the coarse LES mesh but containing only one point in the spanwise direction. All the models (both RANS and LES) predicted the location of the separation point to within 12% of the experimental value but the reattachment point was more difficult to predict. For the RANS simulations both the Spalart-Allmaras and k-ε results were significantly in error. The best RANS results were obtained using the SST model. The DES simulation on the coarse mesh performed marginally worse than the RANS Spalart-Allmaras simulation on the same mesh, while the DES simulation on the fine mesh resulted in an overall worse performance than those on the coarse mesh. The values for the simulated separation and re-attachment points are shown in Table 1. While it was found that all the RANS calculations predicted the mean velocity profiles in the equilibrium boundary layer more accurately than the LES simulations, the reverse was true for the non-equilibrium regions. The RANS models under predicted the magnitude of the back flow velocity and showed a slower recovery downstream of the separation, while the velocity field predicted by the LES simulations was more accurate with the error always less than 10%.

Radhakrishnan et al. [51] have since repeated their DES simulations using a stochastic forcing term in the transition region between the quasi-steady near wall RANS flow and the unsteady LES outer flow. The forcing term generated small scale fluctuations that acted as seeds for the development of realistic energy carrying eddies in the LES region. The inclusion of this additional stochastic term resulted in improved agreement with the experimental results for the DES simulations, as can be seen in Table 1.

More recently El-Askary [52] has performed an LES simulation of this experiment in which the effect of the unresolved sub grid scale stresses on the flow is provided by the inherent dissipation in the numerical scheme. The Favre-filtered compressible Navier-Stokes equations are written in curvilinear coordinates and the MILES method is implemented using a second order accurate AUSM (Advective Upstream Splitting Method) formulation. The convective terms are upstream biased using a properly defined cell interface velocity while the pressure term is computed using the sound velocity. The viscous fluxes are discretized using second order accurate central differences. The mesh contained 1.7 million cells and the first node point in the wall normal direction was located at $y^+ = 1.25$. To save time and computational cost a smaller domain was used in which only 60% of the wind tunnel height was simulated and a slip condition was used on the upper surface. It was found that both the mean flow and the Reynolds stress components agreed well with the experimental data. The mean separation point occurred too early however, at approximately $x' = 0.4$, and the reattachment point was delayed to $x' = 1.7$. The early separation point is the same as that found by Wasistho and Squires [48, 49], while the delay in the reattachment point was attributed to the effects of the lower tunnel height and the slip condition on the upper tunnel surface.

Table 1: Position of separation and re-attachment points for various authors and comparison with experimental results

Authors	Simulation Method	Separation point (percentage error)	Re-attachment point (percentage error)
Song et al. [23]	Experiment	0.77	1.36
Svennberg & Fureby [45]	LES – SMG	0.67 (-13.0%)	1.29 (-5.1%)
Svennberg & Fureby [45]	LES - OEEVM	0.66 (-14.3%)	1.36 (0%)
Wasistho & Squires [48]	LES – Dynamic SMG	≈ 0.4 (≈ 48%)	≈ 1.4 (≈ 3%)
Radhakrishnan et al [50]	RANS - SA	0.79 (4.8%)	1.24 (-23.8%)
Radhakrishnan et al [50]	RANS k-ε	0.79 (4.8%)	1.16 (-36.5%)
Radhakrishnan et al [50]	RANS - SST	0.69 (-11.1%)	1.36 (-4.7%)
Radhakrishnan et al [50]	DES	0.80 (6.3%)	1.46 (11.1%)
Radhakrishnan et al [50]	DES fine mesh	0.72 (-6.3%)	1.48 (14.2%)
Radhakrishnan et al [51]	DES & stochastic forcing	0.75 (-9.5%)	1.32 (-11.1%)
Radhakrishnan et al [51]	DES fine mesh & stochastic forcing	0.70 (-9.5%)	1.41 (3.25%)
El-Askary [52]	MILES	0.4 (48%)	1.7 (25.0%)

6.3 Simulations using oodlesDST

The three-dimensional computational mesh for the simulation was created from a blockMeshDict file and the OpenFOAM utility blockMesh, as explained in a previous report [43]. The domain covered the same spatial extent as shown in Figure 2. A side view of the mesh in the vicinity of the ramp is shown in Figure 3. The mesh contains a total of 2.4 million hexahedral cells. The maximum aspect ratio is 3.9, maximum non-orthogonality is 32.6 with an average value of 2.43 and the maximum skewness is 0.563. Over the two metre long approach flow the axial cell spacing is 5.3 mm and over the 0.5 metre flat plate downstream of the ramp the axial spacing is 2.8 mm. Over the ramp itself the axial cell width reduces to 1.8 mm. The resolution in the spanwise direction is 4 mm per cell and this is kept uniform over the entire length of the domain. The resolution in the wall normal direction over the upstream flat plate is 1.64 mm per cell and this increases to 1.90 mm per cell over the flat plat downstream of the ramp. This results in a y+ value of approximately 40 just upstream of the ramp. The boundary layer thickness at this location is 25.3 mm and this means that there are approximately 15 cells in the boundary layer. This is a relatively coarse mesh, but was chosen to be representative of the amount of resolution which will be available in future typical applications of oodlesDST.

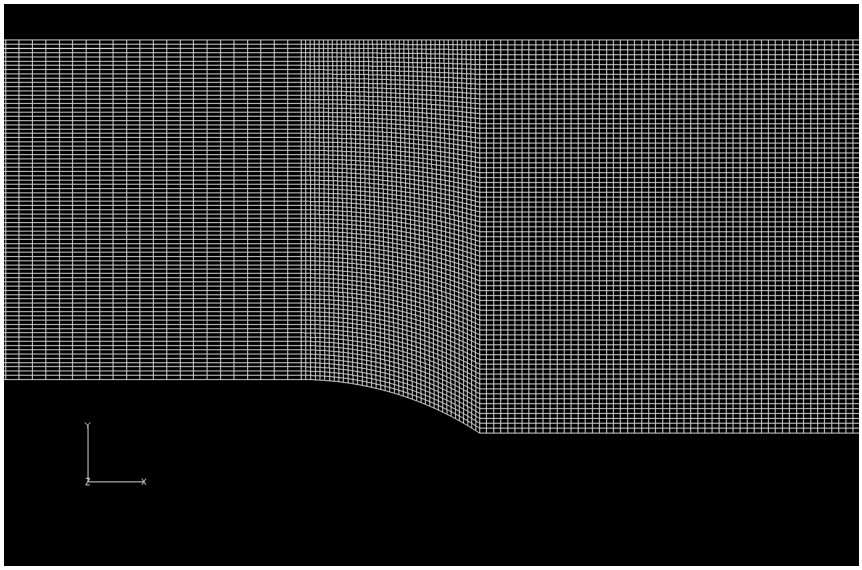

Figure 3: Side view of the computational mesh used to simulate the contoured ramp experiment of Song et al.

Constant values of velocity, subgrid turbulent kinetic energy and viscosity were prescribed on the inflow boundary and zero gradient boundary conditions were applied for these variables on the outflow boundary. For the pressure a zero gradient boundary condition was applied at the inflow boundary and a constant value was specified at the outflow boundary. Zero gradient boundary conditions were used for the pressure and

subgrid turbulent energy on the lower and upper walls while the velocity was set to zero on these boundaries. Symmetry boundary conditions were applied to all variables in the spanwise direction. Since there is a relatively coarse mesh close to the wall Spalding's expression for the law of the wall was used in all simulations.

In the fvSchemes file "linear" discretization was used for the interpolation and gradient schemes and "corrected" for the surface normal gradient scheme. The divergence schemes all used "Gauss linear" except for the ILES simulation which used "Gauss Gamma 0.25". "Gauss linear corrected" was used for all the Laplacian schemes. In the fvSolution file the tolerance was set to 10^{-6} for all variables and the relative tolerance was set to zero. The number of corrector steps and non orthogonal corrector steps in the PISO loop were both set to 2. The time step was set to 3.0×10^{-5} seconds to ensure that the maximum Courant number was kept below 0.5. From a stability point of view this is not strictly necessary as all the time integration schemes in oddlesDST are semi-implicit, however this restriction is required to ensure that the assumptions behind the PISO algorithm are satisfied and that information flows through the grid at a physically realistic rate. The simulations were run for six flow through times to enable initial transients to die out and then statistics were accumulated for a further six flow through times.

All simulations were run in parallel on distributed processors using domain decomposition. The OpenFOAM utility decomposePar is used to split the mesh and fields into a number of sub-domains which are then allocated to separate processors. Communication between processors uses the public domain openMPI implementation of the standard MPI communications protocol. The computer hardware consists of a 128 node cluster based on a Dell M1000e Blade Chassis containing sixteen M610 Blades with dual-Intel Xeon X5560 chips, two M6220 20 port GbE Switches and one M3601Q 32 port Infiniband Switch. The combination of the default openMPI settings in OpenFOAM and the use of Infiniband has been found to limit the scaling of the performance with respect to an increase in the number of processors. Our current implementation reaches peak performance with 48 processors and all simulations have been performed using this number of processors.

Figure 4 shows contours of the instantaneous axial velocity component in the y-z plane at several axial locations for the simulation using the Smagarinsky SGS model. These show the slow growth of the equilibrium boundary layer on the top of the test section and the generation of enhanced turbulence in the boundary layer along the lower surface caused by the flow over the curved ramp.

A better idea of the development of the turbulent boundary layer on the lower surface is shown in Figure 5, where the vortical structures in the flow are identified by plotting iso-contours of the following expression:

$$\sqrt{\nabla(u_x)\cdot\nabla(u_x)+\nabla(u_y)\cdot\nabla(u_y)+\nabla(u_z)\cdot\nabla(u_z)}-\mathrm{mag}\left[\mathrm{curl}(\mathbf{u})\right] \qquad (46)$$

Note that the turbulent structures in the upper boundary layer have been removed in Figure 5 to allow a better view of the lower boundary layer structure. Equation (46) represents the difference between the rate of strain and the rate of rotation and this expression is equivalent to the Q criterion of Hunt et al. [57] which is used by many authors to identify vortex structures. The particular form of equation (46) has been chosen because this expression is easily constructed in the Fieldview software which is used to post-process all our OpenFOAM simulations. The vortical structures are clearly defined when the rate of rotation is greater than the rate of strain and experience has shown that the vortices are best visualized when the above expression has values in the range -100 to -200.

A more detailed view of the vortical structures in the flow after reattachment on the lower plate is shown in Figure 6.

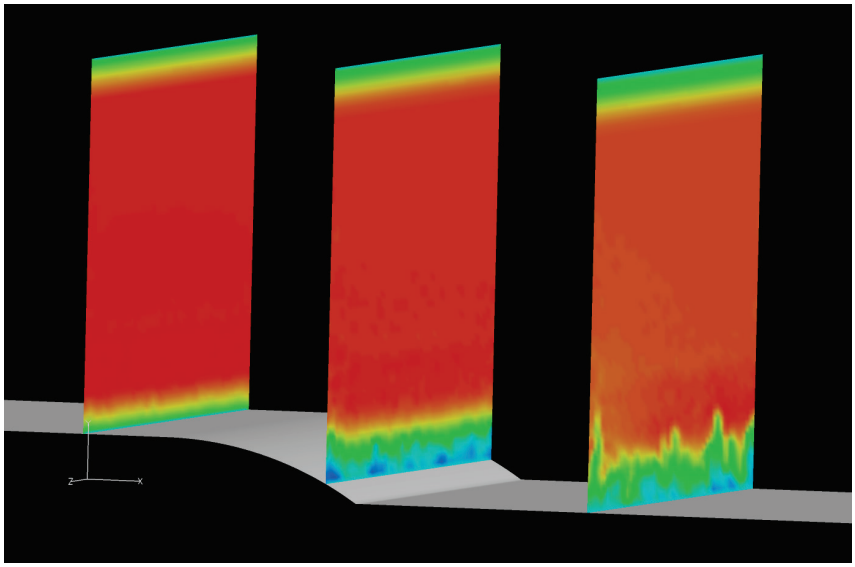

Figure 4: Contours of instantaneous axial velocity component in the y-z plane at several axial locations for the Smagarinsky SGS model.

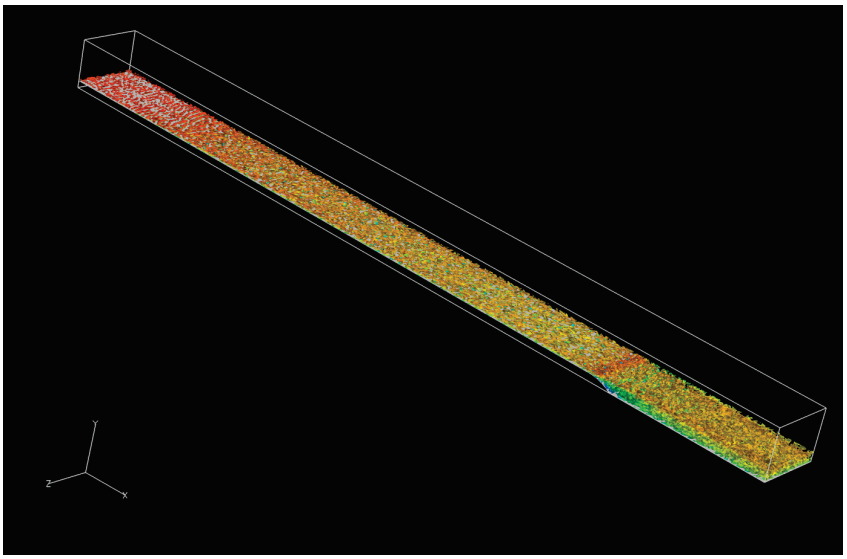

Figure 5: Vortex structure in the lower boundary layer. Iso-contours of equation (46) coloured by the mean value of axial velocity. LDKM model.

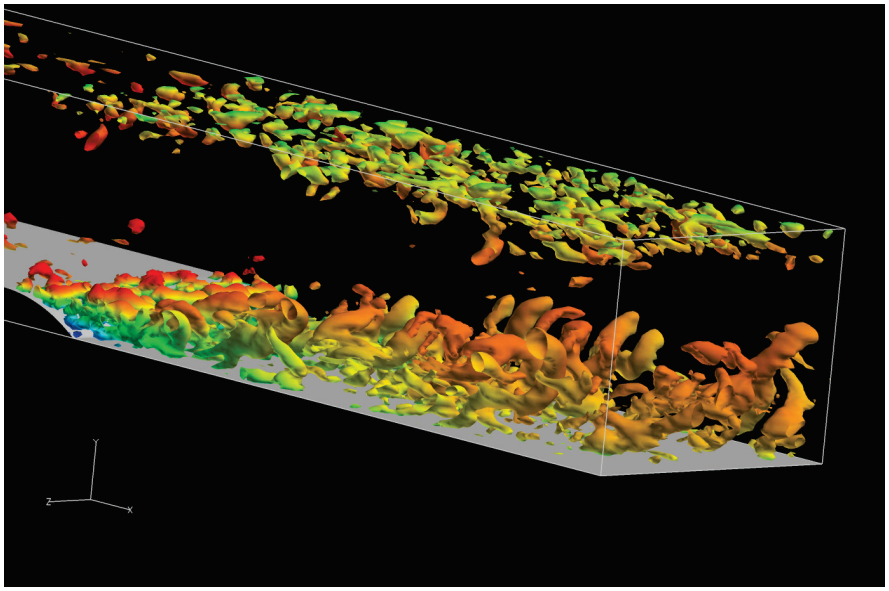

Figure 6: Close-up view of vortical structures in the lower and upper boundary layers. Iso-contours of equation (46) coloured by the mean value of axial velocity. ILES SGS model.

Figures 7 through 11 show a comparison of the experimental streamwise mean velocity profiles at five different axial locations with the simulation results obtained from six different SGS models. In Figure 7 the experimental data shows that an equilibrium turbulent boundary layer has developed along the flat plate upstream of the ramp and each of the SGS models reproduces this behaviour with varying degrees of success. The most accurate is the LDKM model and this is not surprising as it is the most sophisticated of the one equation eddy viscosity models. In Figure 8 the flow has just reached the start of the ramp and is beginning to experience a favourable pressure gradient due to the curvature of the ramp. The LDKM model again provides the best fit to the experimental data. In Figure 9 the flow is starting to separate due to the strong adverse pressure gradient caused by the flow expansion and the experimental data shows a region of negative velocity. This is reproduced to some extent by the ILES, WALE, SMG and OEEVM models, but not by the LDKM and OEEVMMIXED models. In Figure 10 the axial station is located in the middle of the separation bubble and all models show a region of negative velocity. The ILES and WALE models are the best at capturing this reversal in flow velocity while the LDKM model, surprisingly, seems to be the worst. In Figure 11 the flow has moved four ramp lengths along the flat plate at the end of the ramp and the boundary layer has almost completely recovered an equilibrium zero pressure gradient profile. All SGS models show reasonable agreement with the data at this point.

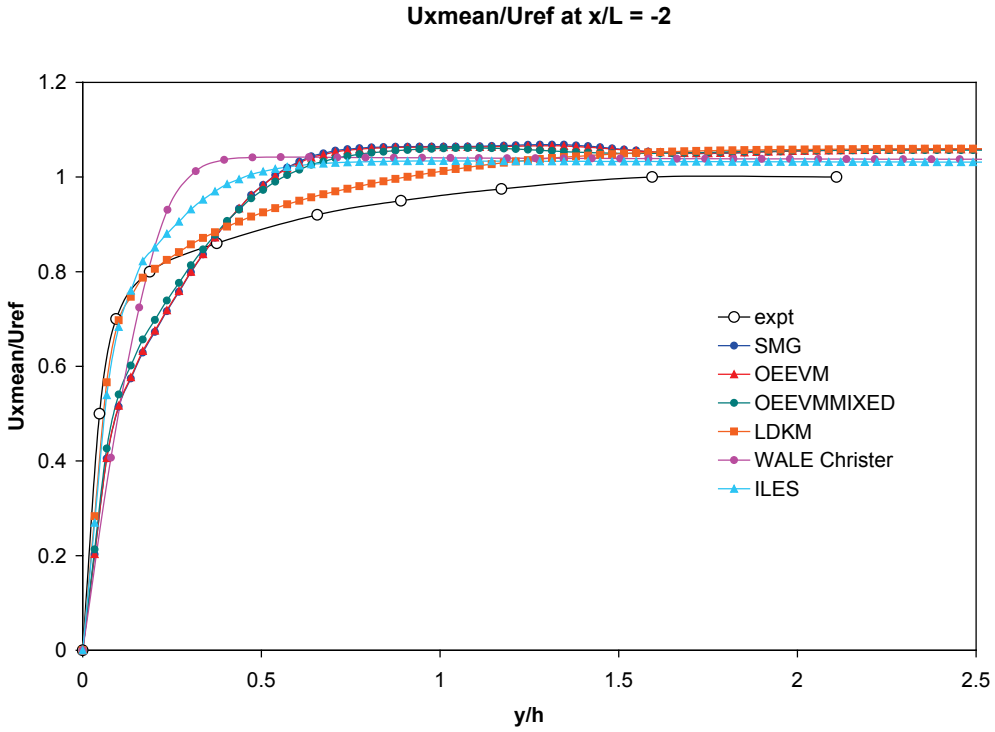

Figure 7: Streamwise mean velocity two ramp lengths upstream of the start of the ramp

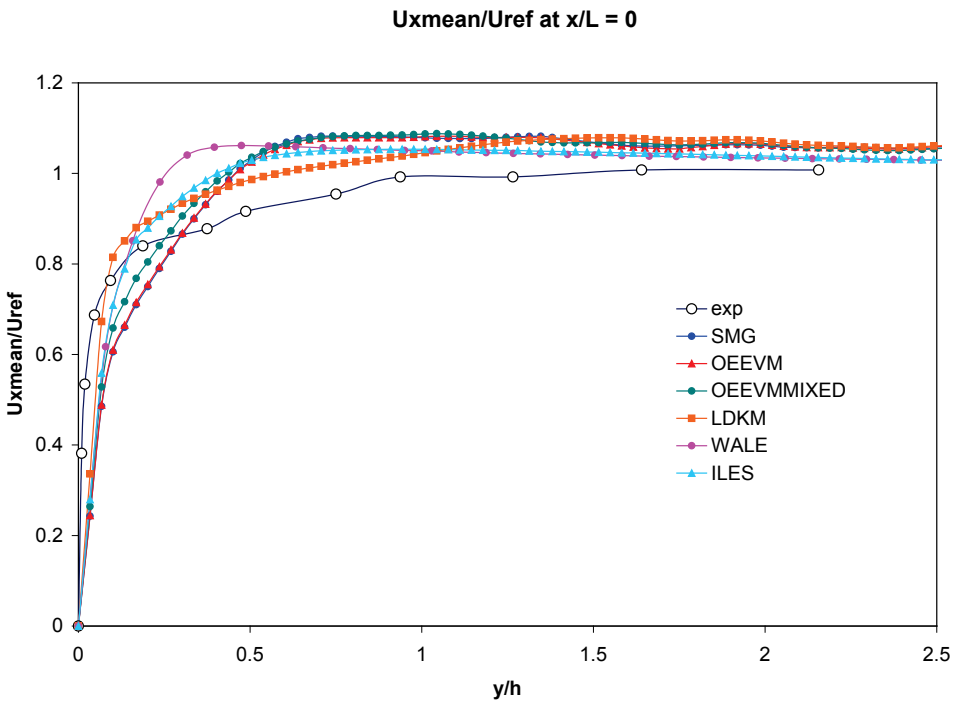

Figure 8: Streamwise mean velocity at the start of the ramp

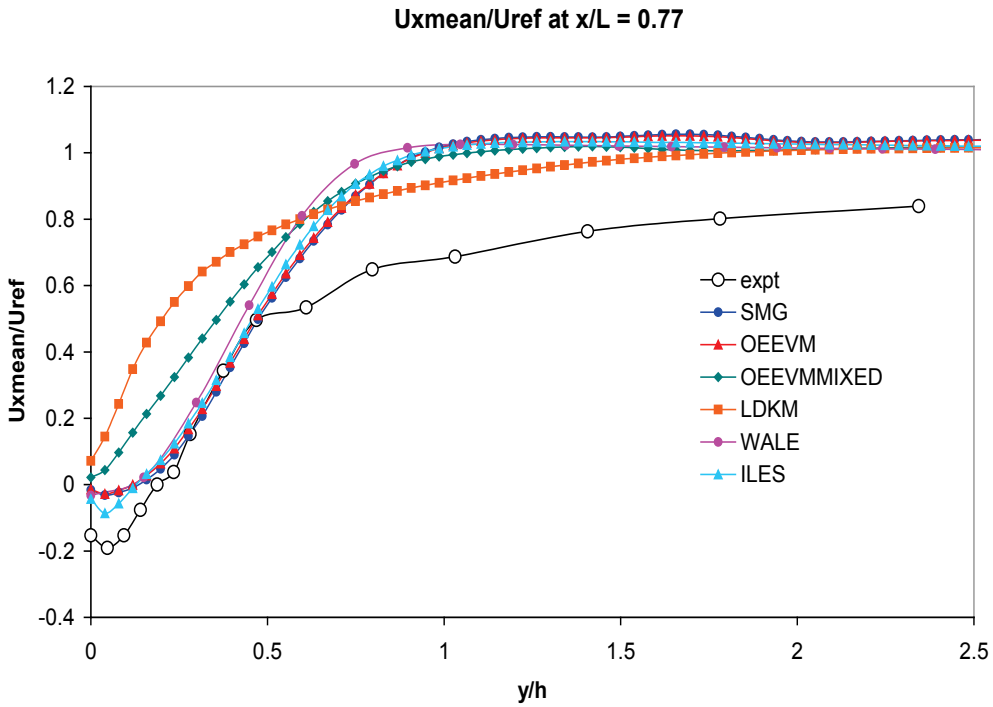

Figure 9: Streamwise mean velocity at the separation point

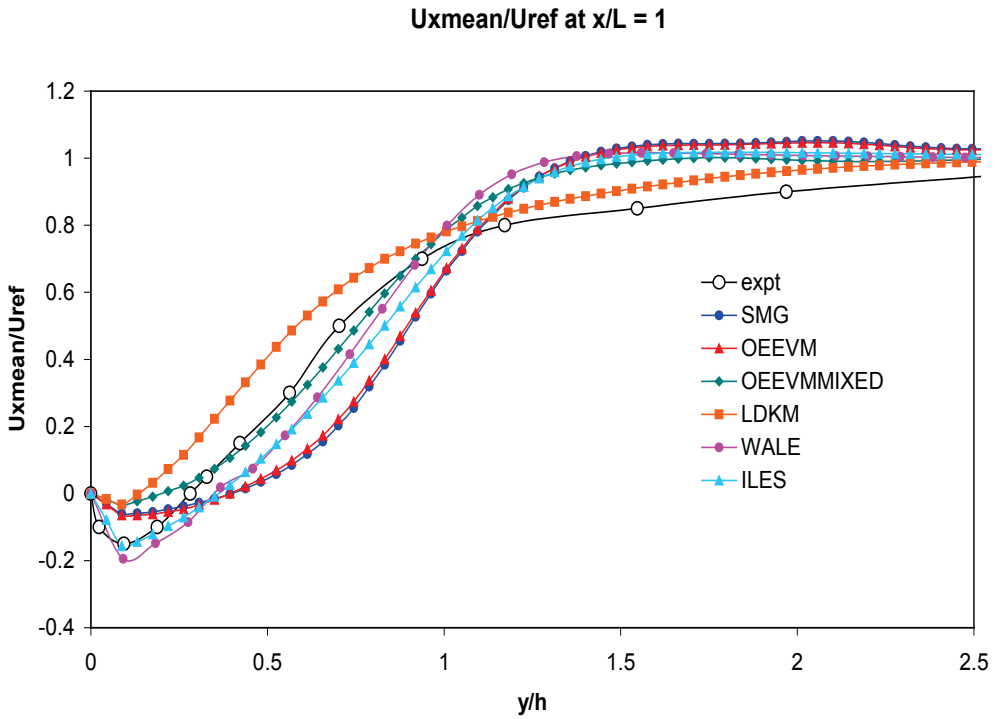

Figure 10: Streamwise mean velocity in the middle of the separation point

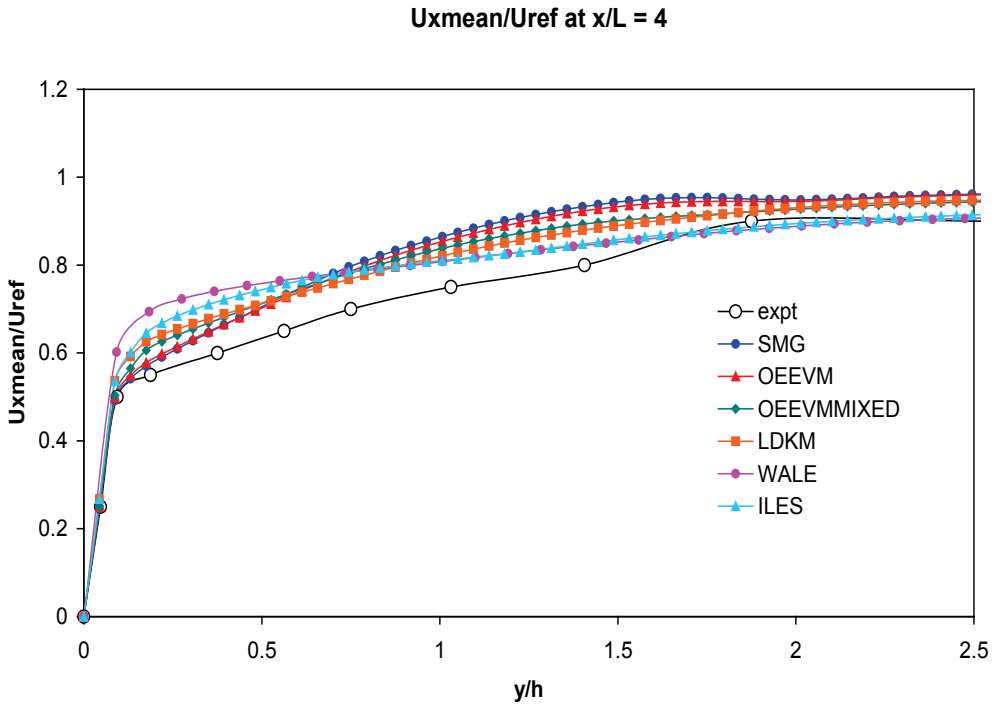

Figure 11: Streamwise mean velocity four ramp lengths downstream of the ramp

One surprising feature of these results is how closely the streamwise mean velocities from the SMG and OEEVM models agree with one another at each of the axial stations. This is not surprising for equilibrium boundary layers as Fureby et al. [16] have recently shown that for this particular case the OEEVM reduces to the SMG model. The strong agreement between the models even in regions of separated flow is unexpected however, although similar agreement between the two models has also been observed by Sidebottom et al. [58]. The poor behaviour of the LDKM model in the separated region is also puzzling, given that Feymark et al [46] and Fureby [47] found that best agreement with the experimental data was obtained with the LDKM model. In their simulation however the LDKM model was used without a wall model and the y^+ values were considerably lower.

Table 2 compares the location of the separation and re-attachment points for each of the SGS models with the experimental values. The simulated values were obtained using the feature abstraction capability of the Fieldview post processing software based on the mean velocity values. The SMG and OEEVM show very similar values for the positions of the separation and re-attachment lines. This is not surprising given the similarity displayed by these two models in the axial velocity plots. The separation point is reasonably well simulated by both models, the error being of the order of 10%, but both models overestimate the re-attachment point with an approximate error of 36%. Svennberg and Fureby [45] studied both of these models on a mesh with a smaller number of cells but better wall resolution. In their simulations the separation point was in error by 13% and 14% for the SMG and OEEVM models respectively, while the re-attachment point for the SMG model was in error by only 5% and the OEEVM predicted the location of the re-attachment point exactly.

Table 2: Position of separation and re-attachment points for each of the SGS models studied and comparison with experimental results.

	Separation point (percentage error)	Re-attachment point (percentage error)
Experiments	0.77	1.36
SMG	0.69 (-10.4%)	1.85 (36.0%)
OEEVM	0.70 (-9.1%)	1.86 (36.8%)
OEEVMMIXED	0.87 (13.0%)	1.30 (-4.4%)
ILES	0.55 (-28.6%)	1.46 (7.4%)
LDKM	0.89 (15.6%)	1.10 (-19.1%)
WALE	0.63 (-18.2%)	1.47 (8.1%)

7. Discussion

The OEEVMMIXED model provides the best overall agreement with the experimental data. The simulated separation point occurs at $x' = 0.87$, which is a 13.0% difference from the experimental value of $x' = 0.77$, and the simulated re-attachment point occurs at $x' = 1.30$, which is a 4.4% error from the experimental value of $x' = 1.36$. This is in agreement with the simulations of Feymark et al. [46] and Furbey [47] who performed simulations on meshes containing up to 8 million cells. The LDKM and WALE SGS models were used on a wall resolved mesh and the LDKM gave best overall agreement. The OEEVMMIXED SGS model plus wall model however gave better agreement than the WALE model on the wall resolved mesh. The OEEVMMIXED did under predict the length of the recirculation bubble, as was the case found here.

The ILES model captures the region of flow reversal better than any of the other SGS models, as can be seen in Figures 6 and 7, however it does separate far too early at $x' = 0.55$ compared with the experimental value of $x' = 0.77$. The re-attachment point is slightly delayed but is only in error by 7.4%. These results are significantly better than the MILES simulation of El-Askary [52], who performed simulations on a grid with a comparable number of mesh points but with a minimum y^+ value of 1.25. In this simulation the separation point occurred far too early at $x' = 0.4$ and the reattachment point was delayed to $x' = 1.7$.

The WALE model also captures the region of flow reversal quite well and the fit to the velocity data in the separated region is good. It also provides reasonable estimates of both the separation point, $x' = 0.63$ (18.2% error) and the reattachment point, $x' = 1.47$ (8.1% error). Overall the performance of this model is quite good and only slightly below that of the OEEVMMIXED.

The LDKM model is the best of the six SGS models at simulating the equilibrium boundary layer on the upstream plate. It is the worst however at capturing the region of reversed flow. The separation point is delayed by 15.6% ($x' = 0.89$ compared with $x' = 0.77$) and the re-attachment point occurs too early ($x' = 1.10$ compared with $x' = 1.36$), resulting in a separation bubble which is much too small. The poor performance of this model, when compared with the excellent results found by Feymark et al [46] and Fureby [47], may be due to the combination of a coarser mesh and the use of the wall model.

Of the six SGS models studied, the LDKM model was found to provide the best agreement with experiment for the equilibrium boundary layer on the upstream flat plate. In the region of separated flow however the LDKM model performed poorly both with respect to the shape of the velocity profiles and the positions of the separation and reattachment points. The OEEVMMIXED model gave the best agreement with experiment for the positions of the separation and reattachment points, while both the ILES and WALE simulations were best at capturing the region of flow reversal. The failure of the LDKM model in this region is surprising given the conclusions of Feymark et al. [46] and Fureby [47], but those simulations were performed on wall resolving meshes. The relatively poor

performance of the LDKM model here may be due to the relatively coarse mesh and the use of a wall model. It would be instructive to repeat the LDKM simulation on a more refined mesh to see if this resulted in improved agreement with the experimental results.

The close agreement found between the results obtained for the mean axial velocity using the SMG and OEEVM models in the separated region warrants further study. As noted by Fureby et al. [57], the OEEVM reduces to the SMG model for equilibrium flows, but the models are not expected to agree in the separated region. Similar agreement between the two models for the mean velocities was also observed by Sidebottom et al. [58] but the models showed differences in the simulated streamwise Reynolds stresses. Since experimental data for the Reynolds stresses exist for the curved ramp experiment, it would be interesting to perform further analysis on the SMG and OEEVM simulation results to see if these differences were observed here.

Experimental data for the skin friction coefficient is available for the curved ramp experiment but has not been considered here because the oodlesDST code currently does not calculate a mean value for the wall shear stress. It would be a simple matter to modify the code to calculate this quantity however and this should be done for future applications of the code.

8. Conclusion

This report has provided a brief introduction to the concept of Large Eddy Simulation of turbulent fluid flow using the oodlesDST software provided to the MD Hydrodynamics Group by the Swedish Defence Research Agency FOI. A detailed description of each of the SGS models and wall-models available in the code has been provided to allow an appropriate choice of these models to be made when running the code. An overview of the code structure has been given and typical solver settings and discretization choices have been illustrated by applying the code to the separation and reattachment of a turbulent boundary layer on a three-dimensional curved ramp. The acquisition of this software now enables the DST Group to perform enhanced simulations of submarine manoeuvres and provides a computational framework for the simulation of acoustic signatures.

9. Acknowledgements

We thank Dr. Mattias Liefvendahl and Dr. Michel Chapuis from FOI for explaining many features of the oodlesDST code during a three day course given at DSTO Fishermans Bend in November 2010. We are grateful to Mr. Brendon Anderson, Head of the Hydrodynamics Group, MPD, for his tireless efforts in setting up this collaboration and for his continued support of this work. We thank Dr. Darrin Stephens from Applied CCM for assistance in understanding some aspects of the open source OpenFOAM coding.

10. References

1. Smagorinsky, J. "General Circulation Experiments with the Primitive Equation. 1. The Basic Experiment", *Mon. Weather Rev.* **91**, 99-164 (1963).

2. Leonard, A. "Energy Cascade in Large-Eddy Simulations of Turbulent Fluid Flows", *Advances in Geophysics* A **18**, 237 -248, (1974).

3. Pope, S.B., "Turbulent Flow", Cambridge University Press, (2000).

4. Nicoud, F and Ducros, F., "Subgrid-scale stress modelling based on the square of the velocity gradient tensor", *Flow, Turbulence and Combustion*, **62**, 183 (1999).

5. Menter, F., "Advanced Turbulence Modelling", August (2012).

6. Lesieur, M., Métais, O. and Comte, P., "Large Eddy Simulations of Turbulence", Cambridge University Press (2005).

7. Oran, E.S. and Boris, J.P., "Numerical Simulation of Reactive Flow", Second Edition, Cambridge University Press (2001).

8. Davidson, P.A., "Turbulence – An Introduction for Scientists and Engineers", Oxford University Press (2004).

9. Deardorff, J.W. "A Numerical Study of Three-Dimensional Turbulent Channel Flow at large Reynolds numbers", *J. Fluid Mech.*, **41**, 453-465 (1970).

10. Germano, M., Piomelli, P.M. and Cabot, W.H., "Dynamic Subgrid Scale Eddy Viscosity Model", *Physics of Fluids A*, **3**, 1760-1765 (1991).

11. Lilly, D.K., "A Proposed Modification of the Germano Subgrid Scale Closure Model", *Physics of Fluids A*, **4**, 633-635 (1992).

12. Sagaut, P., "Turbulence, Direct Numerical Simulation and Large Eddy Simulation", Chapter 9 of "*Encyclopaedia of Computational Mechanics, Volume 3: Fluids*", Edited by E. Stein, R. de Borst and T.J.R Hughes, John Wiley and Sons Ltd. (2004).

13. Bernard, P.S. and Wallace, J.M., "Turbulent Flow: Analysis, Measurement and Prediction", John Wiley and Sons Ltd, Hoboken, N.J. (2002).

14. Schumann, U., "Subgrid Scale Model for Finite Difference Simulation of Turbulent Flows in Plane Channels and Annuli", *J. Comp. Phys.*, **18**, 376 (1975).

15. Yoshizawa, A. and Horiuti, K., "A Statistically-Derived Subgrid Scale Kinetic Energy Model for Large Eddy Simulation of Turbulent Flows", *J. Phys. Soc. Japan*, **54**, 2834 (1985).

16. Fureby, C., Chapuis, M. and Liefvendahl, M., "LES Past the Bare and Fully Appended Joubert Hull", Draft FOI Report, October (2011).

17. Bardina, J., Ferziger, J.H. and Reynolds, W.C., "Improved Subgrid Scale Models for Large Eddy Simulation", AIAA 13th Fluid and Plasma Dynamics Conference, Snowmass, Colorado, AIAA-80-1357 (1980).

18. Liu, S., Meneveu, C. and Katz, J., "On the Properties of Similarity Subgrid-scale Models as deduced from Measurements in a Turbulent Jet", *J. Fluid Mech.*, **275**, pp. 83-119 (1994).

19. Kim, W.W. and Menon, S., "A New Dynamic One-Equation Sub-Grid Scale Model for Large Eddy Simulations", *AIAA paper 95-0356* (1995).

20. Kim, W.W. and Menon, S., "Application of the Localized Dynamic Subgrid-Scale Model to Turbulent Wall-Bounded Flows", *AIAA paper 97-0210* (1997).

21. Deardorff, J. W., "The Use of Subgrid Transport Equations in a Three-Dimensional Model of Atmospheric Turbulence", *Trans. ASME J. Fluids Eng.*, **95**, 429 (1973).

22. Fureby, C., Tabor, G., Weller, H. G. and Gosman, A. D., "Differential Subgrid Stress Models in Large Eddy Simulations", *Phys. Fluids* **9**, 3578-3580 (1997).

23. Song, S., DeGraaff, D.B. and Eaton, J.K., "Experimental Study of a Separating, Reattaching, and Redeveloping flow over a Smoothly Contoured Ramp". *International Journal of Heat and Fluid Flow*, **21**, 512-519 (2000).

24. Boris, J.P., "On Large-Eddy Simulation using Subgrid Turbulence Models", in *Whither Turbulence? Turbulence at the Crossroads*, ed. J.L. Lumley, New York: Springer-Verlag, 344-353 (1990).

25. Boris, J.P. and Book, D.L., "Flux-Corrected Transport 1: SHASTA – A Fluid Transport Algorithm that works", *Journal of Computational Physics*, **11**, 38-69 (1973).

26. Boris, J.P., Grinstein, F.F., Oran, E.S. and Kolbe, R.L., "New Insights into Large Eddy Simulation", Naval Research Laboratory, Washington, DC, NRL/MR/4400-92-6979, April 30 (1992).

27. Fureby, C. and Grinstein, F.F., "Large Eddy Simulation of High-Reynolds-Number Free and Wall-Bounded Flows", *Journal of Computational Physics* **181**, 68-97 (2002).

28. Grinstein, F.F. and Fureby, C., "From Canonical to Complex Flows: Recent Progress on Monotonically Integrated LES", IEEE *Computing in Science and Engineering* **6**, 36-49 (2004).

29. Grinstein, F.F, Margolin, L.G. and Rider, W.J., "Implicit Large Eddy Simulation – Computing Turbulent Fluid Dynamics", Cambridge University Press (2007).

30. Jasak, H., "Error Analysis and Estimation for the Finite Volume Method with Application to Fluid Flows, Ph.D. thesis, Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine, June (1996).

31. Van Leer, B., "Towards the Ultimate Conservative Difference Scheme III. Upstream-Centred Finite-Difference schemes for Ideal Compressible flow", *J. Comp. Phys.* **23** (3): 263–275 (1977).

32. de Villiers, E., "The Potential of Large Eddy Simulation for the Modelling of Wall Bounded Flows", Ph.D. thesis, Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine, London University, July (2006).

33. Bagget, J.S., Jiménez, J. and Kravchenko, A.G., "Resolution Requirements in Large Eddy Simulations of Shear Flows", *Ann. Res. Briefs, Centre for Turbulence Research*, Stanford University (1997).

34. Fureby, C., Alin, N., Wikström, N., Svanstedt, N., Menon, S. and Persson, L., "On Large Eddy Simulation of High Reynolds Number Wall-Bounded Flows", AIAA 2003-0066, *41st AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV*, (2003).

35. Fureby, C. "On LES and DES of Wall Bounded Flows", *Ercoftac Bulletin* No. 72, March Issue, (2007).

36. Liefvendahl, M., Fureby, C. and Perrson, T., "LES and DES of high Reynolds Number Wall Bounded Flows", AIAA 2006-904, pp 10804 – 10818 (2006).

37. Alin, A., Bensow, R.E., Fureby, C., Huuva, T., and Svennberg, U., "Current Capabilities of DES and LES for Submarines at Straight Course", *Journal of Ship Research*, **54** (3), pp. 184 – 196 (2010).

38. Cabot, W. and Moin, P., "Approximate Wall Boundary Conditions in the Large Eddy Simulation of High Reynolds Number Flow", Turb. Flow and Comb., **63**, 269 (2000).

39. Manhart, M., Peller, N. and Brun, C., "Near-wall Scaling for Turbulent Boundary-layers with Adverse Pressure Gradient", *Theoretical Computational Fluid Dynamics*, **22**, 243 (2008).

40. Spalding, D.B., "A Single Formula for the Law of the Wall", *Trans. ASME, J. Appl. Mech.*, **28**, 455 (1961).

41. Duprat, C., Balarac, G., Métais, O., Congedo, P.M. and Brugière, O., "A Wall-Layer Model for Large-Eddy Simulations of Turbulent flows with/out Pressure Gradient", *Physics of Fluids*, **23**, 015101 (2011).

42. Weller, H.G., Tabor, G., Jasak, H. And Fureby, C.,"A Tensorial Approach to Computational Continuum Mechanics using Object-Oriented Techniques", *Computers in Physics*, **16**, No. 6, pp. 620 – 631 (1998).

43. "OpenFOAM: The Open Source CFD Toolbox", User Guide, Version 2.2.0, available at http://www.openfoam.org/docs/ (2013).

44. Issa, R.L., "Solution of the Implicitly Discretized Fluid Flow Equations by Operator-Splitting", *Journal of Computational Physics*, **62**, pp. 40 – 65 (1986).

45. Svennberg, S.U. and Fureby, C., "LES Computation of Flow over a Smoothly Contoured Ramp", *41st Aerospace Sciences Meeting and Exhibition*, 6-9 January, Reno, Nevada, AIAA 2003-965 (2003).

46. Feymark, A., Chapuis, M., Fureby, C. and Liefvendahl, M., "Large Eddy Simulation of High Re Number Partially Separated Flow", *50th Aerospace Science Meeting*, 9-12 January, Nashville, Tennessee, AIAA 2012-0100 (2012).

47. Fureby, C. "Large Eddy Simulation: A Useful Tool for Engineering Fluid Dynamics", *18th Australasian Fluid Mechanics Conference*, Launceston, Australia, 3-7 December (2012).

48. Wasistho, B. and Squires, K.D., "Numerical Investigation of the Separated Flow over a Smoothly Contoured Ramp", *Proceedings of the 2nd International Symposium on Turbulence and Shear Flow Phenomena*, Stockholm, Sweden, June 27-29, Vol. 3, pp. 405-410 (2001).

49. Wasistho, B. and Squires, K.D., "Prediction of Turbulent Separation over a Backward-Facing Smooth Ramp", *Journal of Turbulence*, 6, No.1, 1-26 (2005).

50. Radhakrishnan, S., Piomelli, U., Keating, A. and Lopes, A.S., "Reynolds-Averaged and Large-Eddy Simulations of Turbulent Non-Equilibrium flows", *Journal of Turbulence*, **7**, No. 63, 1-30 (2006).

51. Radhakrishnan, S., Keating, A., Piomelli, U. and Lopes, A.S., "Large-eddy simulations of high Reynolds-number flow over a contoured ramp", 44th AIAA Aerospace Sciences Meeting and Exhibit, AIAA 2006-0899 (2006).

52. El-Askary, W.A., "Turbulent Boundary Layer Structure of Flow over a Smooth-Curved Ramp", *Computers and Fluids*, **38**, 1718-1730, (2009).

53. Spalart, P.R. and Allmaras, S.R., "A One –Equation Turbulence Model for Aerodynamic Flows", *La Recherche Aerospatiale*, **1**, 5-21 (1994).

54. Durbin, P.A., "Near-Wall Turbulence Closure without Damping Functions", *Theoretical and Computational Fluid Dynamics*, **3**, 1-13 (1991).

55. Wilcox, D.C., "Turbulence Modelling for CFD", 2nd edition, DCW Industries, Inc. (1998).

56. Menter, F.R., "Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications", *AIAA Journal*, **32**, 1598-1605 (1994).

57. Hunt, J.C.R., Wray, A.A. and Moin, P., "Eddies, Stream and Convergence Zones in Turbulent Flows". *Center for Turbulence Research, Stanford University*. Report Number CTR-S88 (1988).

58. Sidebottom, W.T., Ooi, A and Jones, D., "Large-Eddy Simulation of Flow past a Circular Cylinder at Re_D = 3900", presented at the 18th Australasian Fluid Mechanics Conference, Launceston, 3 – 7 December (2012).

Appendix A: Files contained in oodlesDST.C

- **oodlesDST.C** This the top level code which contains the time loop and the PISO loop (explained below).
- **createFields.H** This section of code creates all the relevant flow variables for the simulation and is an essential component of any OpenFOAM code. For oddlesDSTO the relevant variables are: pressure "p", velocity vector "**U**", subgrid scale kinetic energy "k", effective viscosity "nuEff", subgrid scale stress tensor "**B**", wall y^+ value "yPlus", wall shear stress "wallShearStress", friction velocity "uTau", the expression used for the length scale or filter width approximation, "delta", and "eps", which is an estimate of the dissipation term. Depending on the choice of subgrid scale model and delta approximation not all of these variables will be required for a particular simulation.
- **createAverages.H** This section of code creates the variable which will be used to accumulate the statistics for the flow variables. These are: mean velocity "Umean", mean pressure "pMean", mean effective viscosity "nuEffMean", "Txx", Tyy" , "Tzz" and "pPrime2Mean". Txx is the mean value of U_x^2, with similar expressions for Tyy and Tzz. pPrime2Mean is defined as (p^2)mean-$(\text{pMean})^2$. Both Txx (and related y and z components) and pPrime2Mean are useful quantities which can be used to check the convergence of the second order statistics.
- **calculateAverages.H** This where the averaged variables defined above are actually calculated.
- **writeNaveragingSteps.H** This file controls the output of the averaged variables.
- **readTransportProperties.H** This part of the code reads the transportProperties file which is kept in the "constant" directory. This where the user specifies the choice of SGS model, the type of wall model used, the type of delta approximation used, and also the values of the density and the viscosity. Note that the value of the density is not used in oodlesDST. Its inclusion in the code is left over from previous compressible versions of the code and it has been left in oddlesDST for continuity with oodlesDSTAcoustics.
- **subGridModel.H** This file contains all the coding to calculate each of the SGS models explained in Section 3.
- **wallViscosity.H** This file contains the coding to calculate the wall models explained in Section 4.
- **UEqn.H** This file constructs the appropriate form of the momentum equation, depending on the particular SGS model chosen.
- **pEqn.H** This section of the code solves the Laplacian equation for the pressure.
- **mean.H** This file contains coding to evaluate the variable averages over neighbouring cells required for the Mixed Models and the LDKM model.

Appendix B: Run time loop in oodlesDST.C

```
    Info<< "\nStarting time loop\n" << endl;
    for (runTime++; !runTime.end(); runTime++)
    {
        Info<< "Time = " << runTime.timeName() << nl << endl;
#       include "readPISOControls.H"
#       include "CourantNo.H"
#       include "subGridModel.H"
#       include "wallViscosity.H"
#       include "UEqn.H"
        // --- PISO loop
        for (int corr=0; corr<nCorr; corr++)
        {
            volScalarField rUA = 1.0/UEqn.A();
            U = rUA*UEqn.H();
            phi = ( fvc::interpolate(U) & mesh.Sf() ) + fvc::ddtPhiCorr(rUA, U, phi);
            adjustPhi(phi, U, p);
            for (int nonOrth=0; nonOrth<=nNonOrthCorr; nonOrth++)
            {
#               include "pEqn.H"
                if (nonOrth == nNonOrthCorr)
                {
                    phi -= pEqn.flux();
                }
            }
#           include "continuityErrs.H"
            U -= rUA*fvc::grad(p);
            U.correctBoundaryConditions();
        }
        delete UEqnPtr;
#       include "calculateAverages.H"
        runTime.write();
#       include "writeNaveragingSteps.H"
        Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
            << "  ClockTime = " << runTime.elapsedClockTime() << " s"
            << nl << endl;
    }
    Info<< "End\n" << endl;
    return(0);
}
```

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
DOCUMENT CONTROL DATA

1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)

2. TITLE

Large Eddy Simulations using oodlesDST

3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION)

Document	(U)
Title	(U)
Abstract	(U)

4. AUTHOR(S)

D.A. Jones, C. Fureby, D. Norrison, C. Troeng and R. Widjaja

5. CORPORATE AUTHOR

Defence Science and Technology Group
506 Lorimer St
Fishermans Bend Victoria 3207 Australia

6a. DSTO NUMBER	6b. AR NUMBER	6c. TYPE OF REPORT	7. DOCUMENT DATE
DST Group-TR-3205	AR-016-503	Technical Report	January 2016

8. FILE NUMBER	9. TASK NUMBER	10. TASK SPONSOR	11. NO. OF PAGES	12. NO. OF REFERENCES
2013/1067775	07386	FSP	37	58

13. DSTO Publications Repository

http://dspace.dsto.defence.gov.au/dspace/

14. RELEASE AUTHORITY

Chief, Maritime Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SA 5111

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CITATION IN OTHER DOCUMENTS Yes

18. DSTO RESEARCH LIBRARY THESAURUS

Large Eddy Simulation, Computational Fluid Dynamics, OpenFOAM

19. ABSTRACT

The oodlesDST code is based on OpenFOAM software and performs Large Eddy Simulations of turbulent fluid flows. The code contains a number of specialized subgrid scale turbulence models and wall models. This report describes these models in sufficient detail to allow a new user of the code to make informed decisions regarding the use of these models for particular applications. An overview of the code structure is given and typical solver settings and discretization choices have been illustrated by applying the code to the separation and reattachment of a turbulent boundary layer on a three-dimensional curved ramp. The simulated results are compared with experimental results as well as previous numerical simulations.